

TUGAS AKHIR - KI141502

# **PEMBANGKITAN MAZE DAN PENGATURAN MULTI PEMAIN PADA ‘AVIAR’ GAME BERBASIS VIRTUAL REALITY MENGGUNAKAN RECURSIVE DIVISION ALGORITHM**

Anggit Yudhistira  
NRP 5114100065

Dosen Pembimbing  
Dr.Eng Darlis Herumurti, S.Kom.,M.Kom.  
Imam Kuswardayan, S.Kom., M.T.

DEPARTEMEN INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2018









**TUGAS AKHIR – KI141502**

**PEMBANGKITAN *MAZE* DAN  
PENGATURAN MULTI PEMAIN PADA  
'AVIAR' GAME BERBASIS *VIRTUAL  
REALITY* MENGGUNAKAN *RECURSIVE  
DIVISION ALGORITHM***

Anggit Yudhistira  
NRP 5114100065

Dosen Pembimbing  
Dr.Eng Darlis Herumurti, S.Kom., M.Kom.  
Imam Kuswardayan, S.Kom., M.T.

DEPARTEMEN INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2018

*[Halaman ini sengaja dikosongkan]*



**FINAL PROJECT – KI141502**  
**MAZE GENERATION AND MULTIPLAYER**  
**SETTING ON 'AVIAR' GAME BASED ON**  
**VIRTUAL REALITY USING RECURSIVE**  
**DIVISION ALGORITHM**

Anggit Yudhistira  
NRP 5114100065

Advisor  
Dr.Eng Darlis Herumurti, S.Kom., M.Kom.  
Imam Kuswardayan, S.Kom., M.T.

INFORMATICS DEPARTEMENT  
Faculty of Information and Communication Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018

*[Halaman ini sengaja dikosongkan]*



## LEMBAR PENGESAHAN

### **PEMBANGKITAN MAZE DAN PENGATURAN MULTI PEMAIN PADA 'AVIAR' GAME BERBASIS VIRTUAL REALITY MENGGUNAKAN RECURSIVE DIVISION ALGORITHM**

### **TUGAS AKHIR**

Diajukan Guna Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Bidang Studi Interaksi Grafika dan Seni  
Program Studi S-1 Departemen Informatika  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember

Oleh :

**Anggit Yudhistira**

NRP : 05111440000065

Disetujui oleh Dosen Pembimbing Tugas Akhir

Dr.Eng Darlis Herumurti, S.Kom., M.Kom.

NIP: 197712172003121001



(pembimbing 1)

Imam Kuswardayan, S.Kom, M.T.

NIP: 197612152003121001

(pembimbing 2)

**SURABAYA  
JULI 2018**

*[Halaman ini sengaja dikosongkan]*

# **PEMBANGKITAN MAZE DAN PENGATURAN MULTI PEMAIN PADA ‘AVIAR’ GAME BERBASIS VIRTUAL REALITY MENGGUNAKAN RECURSIVE DIVISION ALGORITHM**

Nama Mahasiswa : Anggit Yudhistira  
NRP : 5114100065  
Departemen : Informatika FTIK-ITS  
Dosen Pembimbing I : Dr.Eng Darlis Herumurti, S.Kom., M.Kom.  
Dosen Pembimbing II : Imam Kuswardayan, S.Kom, M.T.

## **ABSTRAK**

*Kemajuan teknologi telah membuat banyak sekali perubahan-perubahan dalam perkembangan aplikasi permainan. Aplikasi permainan terus muncul dengan berbagai tema, salah satunya adalah permainan jenis labirin. Aplikasi permainan dengan tema labirin sudah sangat banyak macamnya, contoh permainan labirin adalah Pacman dan Bomberman. Sudah banyak permainan sekarang yang berubah dari statis menjadi dinamis, Sehingga penyusunannya perlu disesuaikan dengan aturan permainan pada masing-masing permainan.*

*Pada penelitian ini penulis menawarkan sebuah penggunaan algortima penyusunan labirin yang dapat menghasilkan labirin yang berubah-ubah. Agar aplikasi permainan lebih menarik dan memberikan pengalaman yang baru dan nyata, penulis menggunakan teknologi realitas virtual. Selain itu, karena banyaknya generasi sekarang yang sering memainkan game bersama-sama penulis menambahkan fitur multi pemain agar permainan menjadi lebih seru.*

*Hasil dari pengujian ini akan berupa sebuah permainan yang dapat berjalan di perangkat personal computer menggunakan oculus rift. Permainan ini dibangun dengan Unity Versi 2017.3.0f3 dengan bahasa pemrograman C#. Proses pembuatan asset permainan sebagian besar mengambil dari asset store dan internet. Dengan pengujian beta dan kuesioner*

*dapat disimpulkan permainan AVIAR dalam bidang gameplay dan multiplayer telah mengimplementasikan dengan baik, namun perlu ditingkatkan kenyamanan bagi pemain.*

***Kata kunci: Recursive Division, Maze, Multiplayer, Realitas Virtual***

# **GENERATE AND ADJUSTING MULTIPLAYER ‘AVIAR’ GAME BASED ON VIRTUAL REALITY USING RECURSIVE DIVISION ALGORITHM**

Name : Anggit Yudhistira  
NRP : 5114100065  
Major : Informatics-FTIK ITS  
Supervisor I : Dr.Eng Darlis Herumurti, S.Kom., M.Kom.  
Supervisor II : Imam Kuswardayan, S.Kom, M.T.

## ***ABSTRACT***

*Technological advances have made many changes in developing games. Games have appear with many themes, one of those themes is maze. Games with maze theme are variant with the prominent example of Pacman and Bomberman. Many games nowadays have transformed from static to dynamic, but maze games have unique rules thus they need to be adjusted according to the rules of each games.*

*In this research the writer offers usage of maze building algorithm that coul produce an ever changing maze. For the sake of making a game that gave new and immersive experience, the writer used virtual reality technology. The writer also used multiplaye feature because the generation nowadays prefer playing games together.*

*The result of this testing will be a game that is run on personal computer using oculus rift. This game is built using unity version 2017.3.03f with C# programming language. Most of the asset are taken from the asset store and internet. Using beta testing and questionnaire it is concluded that AVIAR has been well implemented in terms of gameplay and multiplayer, but should be improved in terms of comfortability of its player.*

***Keywords: Recursive Division, Maze, Multiplayer, Virtual Reality***

*[Halaman ini sengaja dikosongkan]*

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis masih dikaruniai kesehatan dan keselamatan untuk dapat menyelesaikan Tugas Akhir yang berjudul “PEMBANGKITAN MAZE DAN PENGATURAN MULTI PEMAIN PADA ‘AVIAR’ GAME BERBASIS VIRTUAL REALITY MENGGUNAKAN *RECURSIVE DIVISION ALGORITHM*”.

Pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang sangat baik bagi penulis. Dengan pengerjaan tugas akhir ini penulis bisa mendapatkan ilmu lebih serta menerapkan ilmu-ilmu yang penulis dapatkan semenjak awal perkuliahan sampai hari ini di Departemen Informatika FTIK ITS.

Dalam pelaksanaan dan pembuatan tugas akhir ini, penulis menerima banyak bantuan dari berbagai pihak, oleh karena itu melalui lembaran kata pengantar ini, penulis menyampaikan ucapan terima kasih kepada:

1. Allah SWT serta junjungan besar Nabi Muhammad SAW.
2. Ayah dan Ibu penulis, Darmadi dan Endang Paniti yang tiada henti memberikan doa, moral, dan material kepada penulis.
3. Adik penulis, Berlian Ratri yang telah memberikan semangat kepada penulis.
4. Bapak Dr.Darlis Herumurti, S.Kom., M.Kom. dan Bapak Imam Kuswardayan, S.Kom., M.T. selaku pembimbing I dan pembimbing II yang telah memberikan dukungan, motivasi, nasihat dan bimbingan dalam menyelesaikan Tugas Akhir ini.
5. Bapak Dr.Darlis Herumurti, S.Kom., M.Kom. selaku dosen wali penulis yang memberikan arahan kepada penulis.

6. Bapak dan Ibu Dosen di Departemen Informatika yang telah memberikan ilmu selama penulis kuliah.
7. Seluruh Staf dan karyawan Departemen Informatika yang telah memberikan bantuan selama penulis kuliah.
8. Teman satu kelompok tugas akhir, Aufar Rizqi dan Vinsensia Sipriana Zega yang memberikan masukan dan semangat kepada penulis.
9. Rekan-rekan di laboratorium IGS, Rijal, Panji, Nanda, Deni, RianD, Shafli, Bonar, Wildan, Firman, dan lainnya, yang telah saling mendukung dan menyemangati penulis.
10. Sahabat Penulis Rochman, Adi, Satria, Rizal, Iqbal, Romi, Fauzan, Dika, Yusuf, Evan, Rian, Bagas, Angga, Fito, Adit, Danis yang selalu mendukung dan menghibur penulis.
11. Teman-teman angkatan 2014 yang telah membantu, berbagi ilmu, menjaga kebersamaan, dan memberi motivasi kepada penulis, kakak-kakak angkatan 2013, 2012, serta adik-adik angkatan 2015 dan 2016 yang membuat penulis untuk selalu belajar.
12. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis memohon maaf apabila terdapat kekurangan dalam penulisan Tugas Akhir ini. Kritik dan saran penulis harapkan untuk perbaikan dan pembelajaran di kemudian hari. Semoga Tugas Akhir ini dapat memberikan Manfaat yang sebesar besarnya.

Surabaya, Mei 2018



## DAFTAR ISI

<b>ABSTRAK .....</b>	<b>vii</b>
<b>ABSTRACT .....</b>	<b>ix</b>
<b>KATA PENGANTAR.....</b>	<b>xi</b>
<b>DAFTAR ISI .....</b>	<b>xiii</b>
<b>DAFTAR GAMBAR.....</b>	<b>xvii</b>
<b>DAFTAR TABEL .....</b>	<b>xix</b>
<b>DAFTAR KODE SUMBER .....</b>	<b>xxi</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Batasan Masalah .....	2
1.4. Tujuan .....	2
1.5. Manfaat .....	2
1.6. Metodologi Pembuatan Tugas Akhir.....	3
1.7. Sistematika Penulisan.....	4
<b>BAB II TINJAUAN PUSTAKA .....</b>	<b>7</b>
2.1. Rancang Bangun Perangkat Lunak .....	7
2.2. Human Computer Interaction .....	7
2.3. Unity (Game Engine) .....	7
2.4. Bahasa Pemrograman C# .....	8
2.5. Virtual Reality.....	8
2.6. Visual Studio.....	9

2.7.	Oculus Rift.....	9
2.8.	Multiplayer.....	10
2.9.	Algoritma Recursive Division.....	10
<b>BAB III ANALISIS DAN PERANCANGAN SISTEM.....</b>		<b>13</b>
3.1.	Analisis .....	13
3.1.1.	Deskripsi Game AVIAR .....	13
3.1.2.	Arsitektur Sistem.....	14
3.1.3.	Analisis <i>Gameplay</i> Aturan Permainan.....	14
3.1.4.	Analisis Komponen Permainan .....	15
3.2.	Perancangan Perangkat Lunak .....	16
3.2.1.	Perancangan <i>Asset</i> .....	16
3.2.2.	Perancangan Pembangkit <i>Maze</i> menggunakan Algoritma <i>Recursive Division</i> .....	16
3.2.3.	Perancangan <i>Level</i> .....	21
3.2.4.	Perancangan Pergerakan Karakter Pemain .....	22
3.2.5.	Perancangan Timer dalam Permainan .....	23
3.2.6.	Perancangan <i>Health</i> Pemain.....	24
3.2.7.	Perancangan <i>PowerUp</i> .....	24
3.2.8.	Perancangan Multiplayer.....	24
3.3.	Perancangan Antarmuka .....	24
<b>BAB IV IMPLEMENTASI SISTEM.....</b>		<b>27</b>
4.1.	Lingkungan Pengembangan Sistem .....	27
4.2.	Implementasi Pembuatan Aplikasi .....	27
4.2.1.	Implementasi Import <i>Asset</i> dan <i>Package</i> .....	28
4.2.2.	Implementasi Maze Generator .....	32
4.2.3.	Implementasi <i>Level</i> .....	34
4.2.4.	Implementasi Pergerakan Karakter Pemain.....	35
4.2.5.	Implementasi Interaksi dengan Objek .....	37
4.2.6.	Implementasi <i>Timer</i> dalam Permainan .....	39
4.2.7.	Implementasi <i>Health</i> pemain.....	43
4.2.8.	Implementasi <i>PowerUp</i> .....	44

4.2.9.	Implementasi <i>Multiplayer</i> .....	49
4.3.	Implementasi Antarmuka .....	50
4.3.1.	Implementasi Antarmuka <i>Main Menu</i> .....	50
4.3.2.	Implementasi Antarmuka <i>Level Selection</i> .....	52
4.3.3.	Implementasi Antarmuka <i>Gameplay Stage</i> .....	56
<b>BAB V PENGUJIAN DAN EVALUASI.....</b>		<b>57</b>
5.1.	Lingkungan Pengujian .....	57
5.2.	Pengujian Fungsionalitas .....	57
5.2.1.	Pengujian Pembangkit <i>Maze</i> .....	58
5.2.2.	Pengujian Multiplayer .....	62
5.2.3.	Pengujian Pergerakan Karakter Pemain .....	64
5.2.4.	Pengujian Interaksi dengan Objek .....	65
5.2.5.	Pengujian Menang dan Kalah .....	68
5.2.6.	Pengujian Main Menu .....	70
5.2.7.	Pengujian Level Selection .....	72
5.3.	Pengujian Pengguna .....	73
5.3.1.	Skenario Pengujian Pengguna .....	74
5.3.2.	Daftar Pengujian Pemain .....	78
5.3.3.	Hasil Pengujian Pengguna .....	78
5.3.4.	Kritik dan Saran Pengguna .....	79
5.4.	Evaluasi Pengujian .....	80
<b>BAB VI KESIMPULAN DAN SARAN .....</b>		<b>83</b>
6.1.	Kesimpulan .....	83
6.2.	Saran .....	83
<b>DAFTAR PUSTAKA.....</b>		<b>85</b>
<b>LAMPIRAN .....</b>		<b>87</b>
<b>BIODATA PENULIS .....</b>		<b>117</b>

*[Halaman ini sengaja dikosongkan]*

## DAFTAR GAMBAR

Gambar 2.1 Seperangkat Oculus Rift.....	10
Gambar 3.1 Arsitektur AVIAR .....	14
Gambar 3.2 Diagram Alir <i>Recursive Division</i> .....	17
Gambar 3.3 Diagram Alir untuk Mendapatkan nilai Orientation	18
Gambar 3.4 Simulasi Pemabangkit <i>Maze</i> .....	20
Gambar 3.5 Tombol-tombol pada Oculus Touch .....	23
Gambar 3.6 Rancangan <i>Main Menu</i> .....	25
Gambar 3.7 Rancangan Antarmuka <i>Level Selection</i> .....	25
Gambar 3.8 Rancangan Antarmuka <i>Gameplay</i> .....	26
Gambar 4.1 Implementasi <i>Import Asset</i> .....	28
Gambar 4.2 Objek Dinding .....	29
Gambar 4.3 Objek Pilar .....	29
Gambar 4.4 Objek Lantai .....	30
Gambar 4.5 Objek <i>Powerup Health</i> .....	30
Gambar 4.6 Objek <i>Powerup Double Damage</i> .....	31
Gambar 4.7 Objek <i>Powerup Damage Reduction</i> .....	31
Gambar 4.8 Objek <i>Powerup Speedup</i> .....	32
Gambar 4.9 Objek <i>Finish</i> .....	32
Gambar 4.10 <i>Sphere Collider</i> pada <i>prefab</i> karakter.....	37
Gambar 4.11 <i>Grabber</i> pada <i>handanchor</i> .....	38
Gambar 4.12 Grabbable pada objek .....	38
Gambar 4.13 Labirin Saat Siang .....	40
Gambar 4.14 Labirin Saat Malam .....	40
Gambar 4.15 Memasukan <i>Network Lobby</i> ke <i>scene</i> baru.....	49
Gambar 4.16 Memasukkan <i>lobby scene</i> dan <i>play scene</i> .....	49
Gambar 4.17 Memasukkan objek karakter.....	49
Gambar 4.18 Memasukkan <i>spawnable</i> objek.....	50
Gambar 4.19 Implementasi Antarmuka <i>Main Menu</i> .....	51
Gambar 4.20 Implementasi Antarmuka <i>Level Selection</i> .....	53
Gambar 4.21 Implementasi Antarmuka <i>Gameplay Stage</i> .....	56
Gambar 5.1 a.Bentuk <i>maze level 5</i> pada saat siang hari b.Bentuk <i>maze level 5</i> pada saat malam hari .....	58
Gambar 5.2 Bentuk <i>Maze Percobaan 1</i> .....	60

Gambar 5.3 Bentuk <i>Maze</i> Percobaan 2 .....	60
Gambar 5.4 Bentuk <i>Maze</i> Percobaan 3 .....	61
Gambar 5.5 Bentuk <i>Maze</i> Percobaan 4 .....	61
Gambar 5.6 Bentuk <i>Maze</i> Percobaan 5 .....	62
Gambar 5.7 Tampilan <i>Network Lobby</i> .....	62
Gambar 5.8 Tampilan <i>Network Lobby</i> .....	63
Gambar 5.9 Tampilan <i>input IP</i> pada <i>Client</i> .....	63
Gambar 5.10 Tampilan <i>Client join</i> ke <i>Server</i> .....	64
Gambar 5.11 Tampilan pergerakan karakter dan bertemu <i>finish</i> .....	65
Gambar 5.12 Tampilan Berinteraksi dengan Objek .....	66
Gambar 5.13 Tampilan menemukan <i>PowerUp</i> .....	67
Gambar 5.14 Tampilan Menang .....	68
Gambar 5.15 Tampilan Kalah .....	69
Gambar 5.16 Antarmuka <i>Main Menu</i> .....	70
Gambar 5.17 Tampilan <i>Level Selection</i> .....	71
Gambar 5.18 Pengujian <i>Level Selection</i> .....	73

## DAFTAR TABEL

Tabel 3.1 Konfigurasi <i>Level Game</i> .....	22
Tabel 3.2 Tombol dan Fungsi Oculus Touch pada AVIAR .....	23
Tabel 4.1 Spesifikasi Lingkungan Implementasi .....	27
Tabel 5.1 Lingkungan Pengujian .....	57
Tabel 5.2 Skenario Uji Coba Fungsionalitas .....	58
Tabel 5.3 Hasil Uji Coba Pembangkit <i>Maze</i> .....	58
Tabel 5.4 Hasil Uji Coba <i>Multiplayer</i> .....	64
Tabel 5.5 Hasil Uji Coba Pergerakan Karakter Pemain .....	65
Tabel 5.6 Hasil Uji Coba Interaksi dengan Objek .....	67
Tabel 5.7 Hasil Uji Coba Menang dan Kalah .....	69
Tabel 5.8 Uji Coba <i>Main Menu</i> .....	71
Tabel 5.9 Hasil Uji Coba <i>Level Selection</i> .....	73
Tabel 5.10 Rentang Nilai .....	74
Tabel 5.11 Kuesioner Pengguna .....	75
Tabel 5.12 Daftar Penguji Permainan .....	78
Tabel 5.13 Hasil Pengujian Pengguna .....	79
Tabel 5.14 Hasil Akhir Pengujian Pengguna .....	79
Tabel 5.15 Kritik dan Saran Pengguna .....	80

*[Halaman ini sengaja dikosongkan]*



## DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi <i>Maze Generator</i> .....	34
Kode Sumber 4.2 Implementasi <i>Level</i> .....	35
Kode Sumber 4.3 Implementasi Gerakan Pemain .....	37
Kode Sumber 4.4 Implementasi Siang dan Malam .....	39
Kode Sumber 4.5 Implementasi <i>Timer Random Maze</i> .....	43
Kode Sumber 4.6 Implementasi <i>Health</i> Pemain .....	44
Kode Sumber 4.7 Implementasi <i>PowerUp</i> .....	48
Kode Sumber 4.8 Implementasi <i>Main Menu</i> .....	52
Kode Sumber 4.9 Implementasi Antarmuka <i>Level Selection</i> .....	55

*[Halaman ini sengaja dikosongkan]*

# **BAB I**

## **PENDAHULUAN**

Bab ini membahas garis besar penyusunan tugas akhir yang meliputi latar belakang, tujuan pembuatan, rumusan dan batasan permasalahan, metodologi penyusunan tugas akhir, dan sistematika penulisan.

### **1.1. Latar Belakang**

Di zaman yang maju ini telah banyak penemuan-penemuan teknologi, salah satunya adalah virtual reality. Virtual reality (VR) merupakan sebuah teknologi yang memungkinkan user melakukan simulasi terhadap objek nyata yang mampu membangkitkan suasana tiga dimensi (3D). VR ini dapat dimanfaatkan dalam berbagai bidang seperti game, arsitek, pendidikan, kesehatan, simulasi berkendara, bahkan digunakan dalam bidang militer.

Secara garis besar, virtual reality mengenalkan kepada kita sebuah pengalaman digital yang imersif dan nyata dari teknologi-teknologi sebelumnya. Dengan virtual reality kita bisa merasakan sesuatu tanpa harus langsung terjun di dunia nyata dan kita bisa menghindari resiko yang mungkin terjadi di dunia nyata. Hal ini yang membuat teknologi VR banyak digunakan dalam simulasi.

Selain dalam simulasi, banyak juga permainan / *game* berbasis VR. Hal ini banyak dilakukan agar pemain bisa merasakan langsung dunia yang ada dalam permainan tersebut. Selain bisa berinteraksi dengan lingkungan yang ada dan pemain juga dapat mengeksplorasi dunia di dalamnya. Hal ini memberi sebuah pengalaman yang berbeda dan baru bagi semua orang karena selama ini kita hanya bisa memainkan *game* melalui layar saja. Pemain bisa merasakan langsung ketegangan dan keseruan yang ada. Salah satu permainan yang mengasah kemampuan spasial dan memori adalah permainan labirin.

Labirin merupakan sebuah game puzzle berbentuk kumpulan jalan yang memiliki titik awal / titik masuk dan titik akhir / titik keluar. Tujuan dari permainan ini adalah mencari

jalan untuk mencapai titik akhir. Untuk mencari jalan keluar dari labirin, seseorang harus bisa mengingat jalan yang ia ambil dan membayangkan dirinya dalam labirin tersebut agar tahu posisinya dalam labirin. Labirin sendiri masih terhitung jarang bisa ditemukan karena membutuhkan biaya yang besar dan lahan yang luas untuk membuat sebuah labirin.

### **1.2. Rumusan Masalah**

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana membangkitkan *maze* secara acak menggunakan algoritma *recursive division*?
2. Bagaimana cara mengimplementasikan *multiplayer* pada permainan?
3. Bagaimana cara merancang skenario dan tingkat kesulitan pada permainan ?

### **1.3. Batasan Masalah**

Permasalahan yang dibahas dalam tugas akhir memiliki beberapa batasan, yakni sebagai berikut:

1. Permainan maksimal bisa dimainkan oleh 2 pemain.
2. Aplikasi Permainan yang dibangun berjalan pada *platform* Oculus Rift.
3. Lingkungan pengembangan menggunakan aplikasi Unity 3D Free License dengan bahasa pemrograman C#.

### **1.4. Tujuan**

Tujuan pembuatan tugas akhir ini antara lain:

1. Membuat permainan labirin berbasis *virtual reality*.
2. Membuat permainan labirin dengan pembangkit menggunakan algoritma *recursive division*.
3. Membuat permainan labirin yang dapat dimainkan secara *multiplayer*.

### **1.5. Manfaat**

Manfaat dari tugas akhir ini adalah terciptanya aplikasi

permainan labirin (maze) berbasis Virtual Reality yang dapat dimainkan secara *multiplayer*. Permainan yang dibuat diharapkan dapat memberi solusi terhadap pembangkit permainan dengan lingkungan dinamis dan dapat menghibur pemain.

## 1.6. Metodologi Pembuatan Tugas Akhir

Pembuatan tugas akhir dilakukan menggunakan metodologi sebagai berikut:

### 1. Studi Literatur

Tahap studi literatur merupakan tahap pembelajaran dan pengumpulan informasi yang digunakan untuk mengimplementasikan tugas akhir. Pada tahap studi literatur ini, penulis akan mempelajari beberapa referensi terkait topik tugas akhir. Tahap ini diawali dengan pengumpulan literatur, diskusi, eksplorasi teknologi, dan pustaka, serta pemahaman dasar teori yang digunakan pada topik tugas akhir. Literatur-literatur yang dimaksud disebutkan sebagai berikut:

1. Unity
2. Bahasa pemrograman C#
3. Algoritma Recursive Division

### 2. Analisis dan Desain Perangkat Lunak

Aplikasi ini memiliki fitur-fitur sebagai berikut :

1. Bermain sesuai dengan aturan labirin (*maze*) pada umumnya.
2. *Random* labirin (*maze*) ketika pemain melewati batas waktu.
3. Musuh akan mengejar pemain apabila pemain melewati batas waktu yang sudah ditentukan.
4. *Minigame* di sebuah area tertentu untuk mendapat reward.

### 4. Perancangan Permainan

Pada tahap ini akan dilakukan analisa, perancangan, dan pendefinisian kebutuhan system untuk mengetahui

permasalahan yang akan dihadapi pada tahap implementasi. Kemudian akan dijabarkan kebutuhan-kebutuhan tersebut ke dalam perancangan fitur sistem. Berikut langkah yang akan dilakukan perancangan proses aplikasi:

1. Algoritma *Recursive Division* dalam aplikasi permainan
2. Perancangan *gameplay*
3. Perancangan data dan *asset* permainan.
4. Implementasi dan pembuatan sistem  
Aplikasi ini akan dibangun menggunakan Unity Game Engine versi 5.6.3p2 (64-bit), dengan bahasa pemrograman C#.
5. Pengujian dan Evaluasi  
Tahap pengujian dan evaluasi berisi pengujian aplikasi dan evaluasi berdasarkan hasil pengujian. Pada tahap ini dilakukan pengujian dari fungsionalitas perangkat lunak, apakah sesuai dengan yang diharapkan serta tidak diharapkan terdapat bug.
6. Penyusunan laporan tugas akhir  
Pada tahap ini dilakukan penyusunan laporan yang berisi dasar teori, dokumentasi dari perangkat lunak, dan hasil-hasil implementasi dan pengujian aplikasi perangkat lunak yang diperoleh selama pengerjaan tugas akhir.

### **1.7. Sistematika Penulisan**

Buku Tugas Akhir ini terdiri atas beberapa bab yang tersusun secara sistematis, yaitu sebagai berikut.

1. BAB 1 Pendahuluan  
Menjelaskan latar belakang, batasan masalah, tujuan dari pembuatan tugas akhir ini serta metodologi yang digunakan selama penyusunan.
2. BAB 2 Tinjauan Pustaka  
Memaparkan hasil studi literatur yang digunakan sebagai dasar untuk menyelesaikan tugas akhir ini, terdiri atas deskripsi mengenai perancangan perangkat lunak, human computer interaction, Unity sebagai game engine, bahasa

pemrograman C#, game *maze*, dan algoritma *Recursive Division*.

3. BAB 3 Analisis dan Perancangan Sistem

Pada tahap ini dijelaskan deskripsi dari game AVIAR dan dianalisa bagaimana *gameplay* dari game AVIAR. Setelahnya dibahas mengenai bagaimana perancangan *gameplay*, perancangan stage, dan bagaimana algoritma *Recursive Division* diterapkan dan disesuaikan dengan aturan permainan AVIAR.

4. BAB 4 Implementasi Sistem

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan. Penjelasan implementasi meliputi implementasi pembuatan objek, implementasi pembuatan aplikasi, dan implementasi pembuatan simulasi aplikasi permainan dengan menerapkan algoritma *Recursive Division* yang disesuaikan dengan aturan permainan.

5. BAB 5 Pengujian dan Evaluasi

Pengujian dilakukan dengan menguji setiap stage yang menerapkan algoritma *Recursive Division* dalam penyusunan stagenya. Setiap stage akan diuji coba dan diamati apakah stage dapat diselesaikan atau tidak.

6. BAB 6 Kesimpulan dan Saran

Berisi tentang kesimpulan yang didapat dari proses dan pengujian pembuatan tugas akhir beserta saran-saran untuk pengembangan aplikasi selanjutnya.

*[Halaman ini sengaja dikosongkan]*



## **BAB II**

### **TINJAUAN PUSTAKA**

Bab ini membahas teori-teori yang mendukung pembuatan tugas akhir. Teori yang mendukung tersebut adalah deskripsi mengenai perancangan perangkat lunak, *human computer interaction*, Unity sebagai *game engine*, bahasa pemrograman C#, game *maze*, dan algoritma Recursive Division.

#### **2.1. Rancang Bangun Perangkat Lunak**

Rancang bangun perangkat lunak adalah suatu ilmu yang mempelajari proses pembuatan aplikasi yang melingkupi analisis permasalahan dan kebutuhan, perencanaan, analisis sistem, implementasi, serta pemeliharaan perangkat lunak termasuk perbaikan jika ditemukannya *bug* [1].

#### **2.2. Human Computer Interaction**

Interaksi manusia dan komputer (bahasa Inggris: *human-computer interaction*, HCI) adalah disiplin ilmu yang mempelajari hubungan antara manusia dan komputer yang meliputi perancangan, evaluasi, dan implementasi antarmuka pengguna komputer agar mudah digunakan oleh manusia. Ilmu ini berusaha menemukan cara yang paling efisien untuk merancang pesan elektronik. Sedangkan interaksi manusia dan komputer sendiri adalah serangkaian proses, dialog dan kegiatan yang dilakukan oleh manusia untuk berinteraksi dengan komputer yang keduanya saling memberikan masukan dan umpan balik melalui sebuah antarmuka untuk memperoleh hasil akhir yang diharapkan [2].

#### **2.3. Unity (Game Engine)**

Unity merupakan sebuah game engine yang dikembangkan oleh Unity Technologies. Unity dapat menciptakan game ke dalam beberapa sistem operasi sekaligus. Antara lain: Windows Phone, Android, iOS, Windows 8, OSX, Tizen OS, Blackberry 10, Playstation 3, Playstation 4, XBOX, Oculus Rift dan sebagainya. Game yang dapat dibuat dengan Unity ini bisa dalam bentuk 3D atau 2D [3].

## 2.4. Bahasa Pemrograman C#

Bahasa C# adalah sebuah bahasa pemrograman modern yang bersifat general-purpose, berorientasi objek, yang dapat digunakan untuk membuat program di atas arsitektur Microsoft .NET Framework. Bahasa C# ini memiliki kemiripan dengan bahasa Java, C dan C++ (selengkapnya dapat dilihat pada Sejarah Bahasa C#).

Bahasa pemrograman ini dikembangkan oleh sebuah tim pengembang di Microsoft yang dipimpin oleh Anders Hejlsberg, seorang yang telah lama malang melintang di dunia pengembangan bahasa pemrograman karena memang ialah yang membuat Borland Turbo Pascal, Borland Delphi, dan juga Microsoft J++.

Kini, C# telah distandarisasi oleh European Computer Manufacturer Association (ECMA) dan juga International Organization for Standardization (ISO) dan telah menginjak versi 3.0 yang mendukung beberapa fitur baru semacam Language Integrated Query (LINQ) dan lain-lainnya [4].

## 2.5. Virtual Reality

Realitas virtual atau yang lebih disebut *Virtual Reality* (VR) adalah suatu teknologi yang memberi interaksi grafis secara langsung (*real-time*) dengan model tiga dimensi yang dikombinasikan dengan teknologi layar yang mampu memberi kesan *immersive* (seolah-olah berada pada dunia nyata) kepada pengguna. Realitas Virtual pada dasarnya mengacu pada kesan *immersive*, interaktif, multi-sensor, pengolahan komputer yang menghasilkan lingkungan tiga dimensi dan kombinasi teknologi yang dibutuhkan untuk membangun lingkungan tersebut.

Realitas virtual memungkinkan untuk menyimulasikan dunia nyata ke dalam sistem visualisasi tiga dimensi dan memungkinkan interaksi pengguna dengan model karena kemampuan pelacakan gerak dan kemampuan untuk bertindak dengan produk secara *real time* [5].

## **2.6. Visual Studio**

Microsoft Visual Studio merupakan sebuah aplikasi yang terdiri dari kompiler, SDK, Integrated Development Environment (IDE), dan dokumentasi yang biasa digunakan untuk mengembangkan aplikasi personal, aplikasi web, aplikasi mobile, aplikasi konsol maupun komponen aplikasi pada Microsoft Visual Studio. Banyak pengembang aplikasi yang mengembangkan aplikasinya menggunakan Microsoft Visual Studio karena banyak fungsi yang telah disediakan oleh Microsoft Visual Studio, sehingga pengembang lebih mudah dan sangat terbantu dalam mengembangkan maupun membuat aplikasi [6].

## **2.7. Oculus Rift**

Oculus Rift merupakan sebuah kumpulan perangkat virtual reality yang dikembangkan oleh perusahaan Oculus. Pada tahun 2012, Oculus memulai sebuah kampanye kickstarter. Proyek tersebut menuai kesuksesan dengan mengumpulkan dana sebesar 2,5 juta dollar Amerika [7]. Kemudian pada bulan Maret 2014, Facebook membeli perusahaan Oculus sebesar 2 milyar dollar Amerika. Namun pada tahun 2016, pendiri Oculus memutuskan untuk meninggalkan Facebook.

Oculus Rift terdiri atas beberapa komponen, yaitu Oculus Headset, Oculus Touch, Oculus Sensor, Oculus Remote, Joystick Xbox, dan Connector VR Guitar Controller [8] seperti pada Gambar 2.1.



**Gambar 2.1 Seperangkat Oculus Rift**

(Sumber:

[https://docs.unrealengine.com/portals/0/images/Platforms/Oculus/Rift\\_Touch\\_Tracker.png](https://docs.unrealengine.com/portals/0/images/Platforms/Oculus/Rift_Touch_Tracker.png))

## **2.8. Multiplayer**

*Multiplayer* merupakan sebuah *mode* permainan yang dapat dimainkan lebih dari 1 pemain pada sebuah arena yang sama dan pada waktu yang sama. Pada permainan multiplayer memungkinkan para pemainnya saling berinteraksi dengan pemain lainnya, saling bekerja sama dalam tim, dan saling menjadi lawan yang tidak akan ditemukan pada permainan *singleplayer*. Salah satu *tool* untuk membuat *multiplayer* adalah *network lobby manager*. *Network lobby manager* adalah *network manager* yang menyediakan lobi untuk *multiplayer* yang mudah digunakan sebelum memasuki *scene* utama dalam permainan[9].

## **2.9. Algoritma Recursive Division**

Maze Generator merupakan sebuah metode yang digunakan untuk membuat labirin secara otomatis. Maze generator pada umumnya menggunakan algoritma-algoritma tertentu yang akan menghasilkan labirin secara acak, jadi labirin yang terbentuk tidak akan sama dengan sebelumnya.

Recursive Division merupakan salah satu algoritma dalam maze generator. Algoritma ini bekerja sebagai berikut, diawali

dengan ruang kosong disebut kamar, kemudian pasang dinding secara random baik horisontal maupun vertikal, dan dinding dilubangi secara random agar kamar-kamarnya terhubung. Kemudian rekursif tersebut diulangi secara terus menerus sampai semua subkamar berukuran minimum[10].

*[Halaman ini sengaja dikosongkan]*

## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

Bab ini membahas tahap analisis permasalahan dan perancangan tugas akhir. Pada bagian awal akan dibahas mengenai *gameplay* atau aturan dari game ‘AVIAR’. Selajutnya dibahas mengenai penggunaan algoritma Recursive Division yang disesuaikan dengan aturan permainan.

#### **3.1. Analisis**

##### **3.1.1. Deskripsi Game AVIAR**

AVIAR adalah nama game penulis, nama AVIAR berdasar dari nama-nama penulis yaitu Anggit, Vinsensia, dan Aufar. AVIAR adalah sebuah permainan bergenre labirin/*maze*. Pada permainan AVIAR ini pemain dihadapkan pada sebuah labirin/*maze* dan diharuskan untuk mencari jalan keluar. Bagian yang menarik dalam permainan ini adalah labirin yang bisa berubah-ubah dan musuh yang akan mengejar pemain.

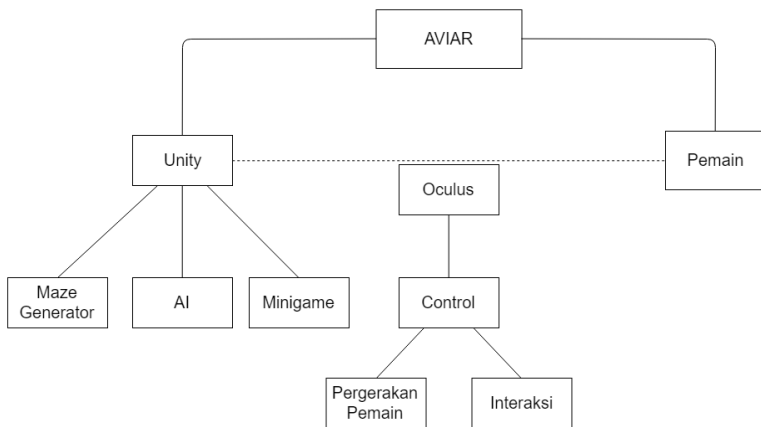
Memang sudah banyak permainan bertema labirin dan juga permainan-permainan yang menggunakan generator sebagai pembangkit gamenya, namun masih jarang sekali dari permainan labirin tersebut yang memberikan pengalaman berbeda. Oleh karena itu, permainan AVIAR ini diciptakan dengan pembangkit *maze* yang dipadukan dengan teknologi *virtual reality* menggunakan Oculus Rift.

Permainan AVIAR ini dibuat dengan Blender untuk membuat objek 3D dan juga Unity karena kedua aplikasi tersebut telah menyediakan fitur yang mendukung Oculus Rift dalam pembuatan *virtual reality*. Permainan ini menerapkan teknologi realitas virtual agar pengguna dapat melihat sekitar dan merasakan pengalaman permainan yang nyata dalam memainkan permainan labirin ini.

### 3.1.2. Arsitektur Sistem

AVIAR merupakan aplikasi permainan VR. AVIAR dibangun menggunakan *tools* Unity3D. Unity3D digunakan untuk membangun *maze generator* yang dikerjakan oleh Anggit Yudhistira, *minigame* yang dikerjakan oleh Aufar Rizqi dan *AI* yang dikerjakan oleh Vinsensia S. Untuk memainkan game AVIAR. Pada permainan AVIAR ini menggunakan oculus touch sebagai kontrol pemain. Kontrol pemain dibagi menjadi 2, yang pertama pergerakan karakter pemain dikerjakan oleh Anggit Yudhistira dan interaksi yang dikerjakan oleh Aufar Rizqi.

Berdasarkan penjelasan diatas, arsitektur dari AVIAR secara sederhana dapat diilustrasikan seperti pada Gambar 3.1 dibawah ini.



**Gambar 3.1 Arsitektur AVIAR**

### 3.1.3. Analisis *Gameplay* Aturan Permainan

AVIAR memiliki aturan yang cukup sederhana. Untuk awal permainan pemain akan dihadapkan pada sebuah labirin. Sama seperti permainan labirin pada umumnya, pemain disuruh mencari jalan keluar didalam labirin untuk menyelesaikannya. Namun, tidak seperti permainan labirin pada umumnya, AVIAR tidak memiliki batas waktu untuk menyelesaikan permainan. Jadi,



pemain bisa selama mungkin berada didalam permainan untuk menyelesaikan permainan.

### **3.1.4. Analisis Komponen Permainan**

Ada beberapa komponen yang ada dalam aplikasi permainan ‘AVIAR’. Komponen tersebut akan mempengaruhi jalannya permainan.

#### **3.1.4.1. Susunan Maze**

AVIAR memiliki susunan *maze*. *Maze* yang ada di dalam permainan ini memiliki luas daerah yang berbeda beda tergantung dari *stage* yang dimainkan. Pada permainan ini pemain pasti mendapatkan susunan *maze* yang berbeda setiap kali memainkan *stage*. Susunan yang berbeda didapatkan dari penggunaan algoritma *Recursive Division*.

#### **3.1.4.2. Musuh**

Musuh adalah salah satu objek dari permainan. Musuh pada permainan AVIAR ini akan bertindak sebagai penghalang/musuh untuk menghambat pemain, sehingga pemain kesusahan mencari jalan keluar untuk menyelesaikan permainan. Musuh ini juga bisa menyerang pemain sehingga pemain bisa terbunuh dan permainan akan berakhir.

#### **3.1.4.3. Waktu siang malam**

Pada permainan AVIAR memang tidak ada batas waktu dalam menyelesaikan game, tetapi ada waktu siang malam sebagai penanda untuk pemain. Setiap pergantian siang ataupun malam bentuk *maze* akan selalu berubah bentuk. Pada saat *nighttime* akan muncul musuh yang akan mengejar pemain. Pemain bisa mengalahkan musuh atau menunggu sampai *daytime*, karena pada waktu siang hari musuh akan menghilang.

#### **3.1.4.4. Minigame**

Didalam permainan AVIAR akan ada beberapa *minigame* yang menyediakan berbagai permainan-permainan sederhana. Ada *chest* disebelah papan *minigame* ini. Pemain bisa

memperoleh hadiah didalam *chest* tersebut apabila bisa menyelesaikan *minigame*.

#### **3.1.4.5. Senjata**

Senjata di AVIAR ini berfungsi untuk mengalahkan musuh pada saat malam hari apabila pemain tidak bisa menyelesaikan permainan selama siang hari. Senjata ini bisa didapat dari chest setelah menyelesaikan *minigame*.

#### **3.1.4.6. Power Up**

*Power Up* didalam permainan AVIAR merupakan objek bantuan untuk pemain. Ada bermacam-macam *power up* dalam AVIAR.

### **3.2. Perancangan Perangkat Lunak**

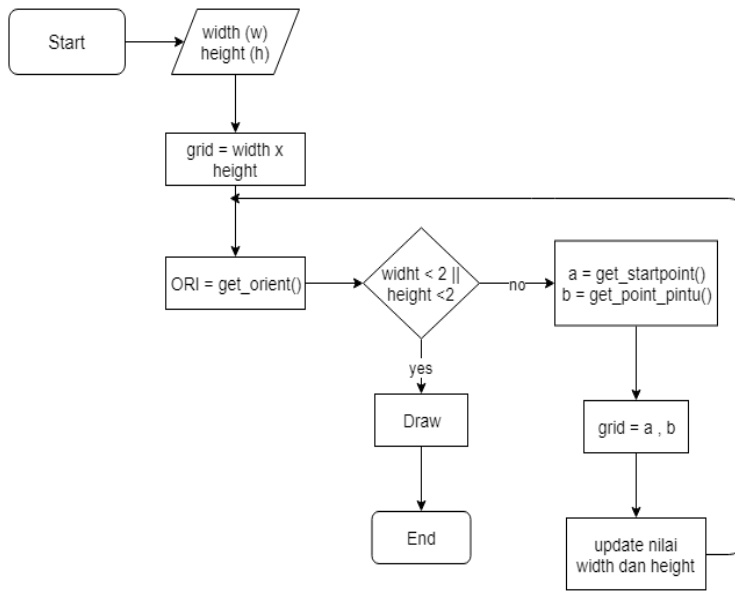
#### **3.2.1. Perancangan Asset**

Pada perancangan aset di permainan AVIAR ini, penulis membedakan aset menjadi 2. Aset internal yang dibuat sendiri oleh penulis dan aset yang didapat dari *asset store*.

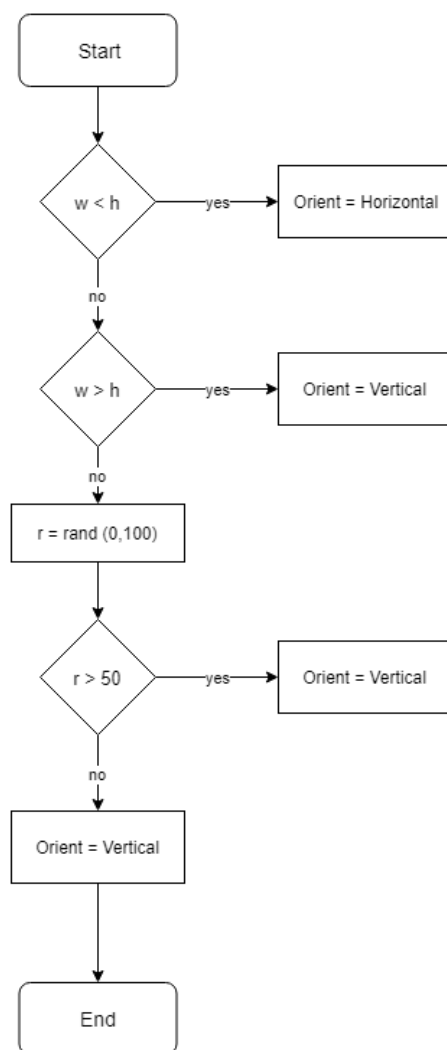
Aset yang dibuat oleh penulis diantaranya adalah tembok, lantai, pilar, karakter pemain, dan *powerup health*. Untuk aset *powerup* dan obor didapatkan penulis dari *asset store*. Kemudian, aset suara didapatkan oleh penulis dari audioblocks.com melalui akun Lab IGS.

#### **3.2.2. Perancangan Pembangkit Maze menggunakan Algoritma Recursive Division**

Perancangan Aplikasi Permainan AVIAR yang berbentuk labirin ini menggunakan *maze generator*. *Maze generator* ini digunakan untuk menciptakan bentuk *maze* secara otomatis dengan pola yang berbeda-beda. Dalam pembuatan *maze generator* ini menggunakan sebuah algoritma, algoritma yang digunakan adalah *Recursive Division* yang diagram alirnya dapat ditunjukkan pada Gambar 3.2 dan Gambar 3.3.



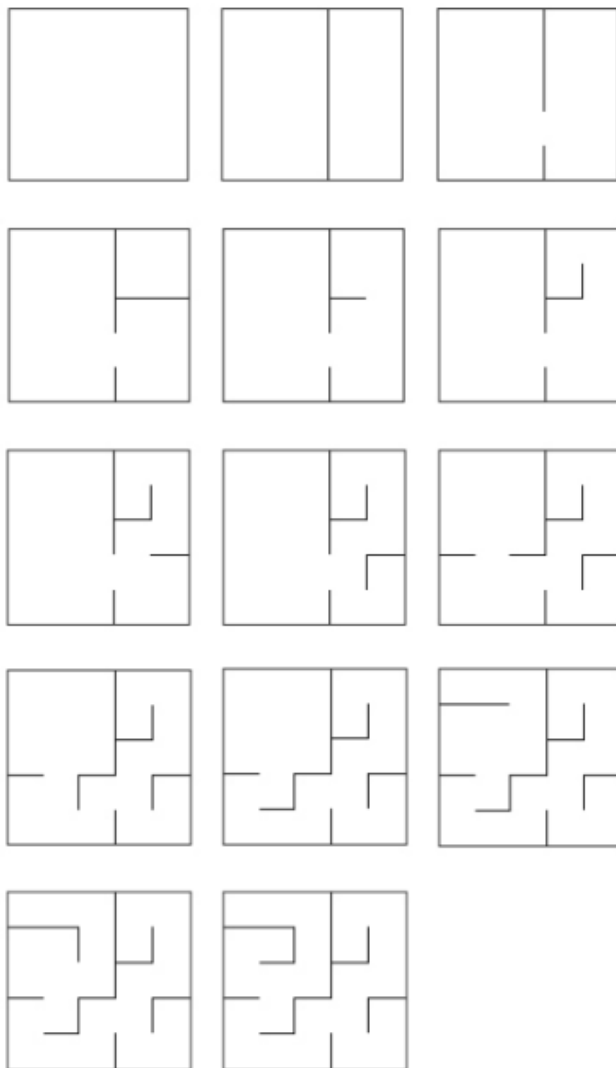
**Gambar 3.2 Diagram Alir *Recursive Division***



**Gambar 3.3 Diagram Alir untuk Mendapatkan nilai Orientation**

Proses pembuatan *maze* menggunakan algoritma *Recursive Division* adalah sebagai berikut:

1. Membuat sebuah area dengan persegi panjang yang disebut ruang.
2. Tambahkan dinding secara vertikal ataupun secara horisontal secara acak kedalam ruang tersebut untuk membagi ruangnya, akan tetapi akan diberikan jalan keluar disalah satu titik secara random pada dinding yang telah dibuat.
3. Ulangi proses seperti langkah ke 2 pada sub ruang yang terbentuk sampai membentuk sebuah *maze* yang sempurna. Proses tersebut berhenti setelah sub ruang yang terbentuk sampai pada saat *cell* terkecil dimana ukuran kolom atau baris adalah 1. Simulasi bisa dilihat pada gambar 3.4.



**Gambar 3.4 Simulasi Pemabangkit *Maze***

Dengan algoritma *Recursive Division* ini maka akan tercipta sebuah labirin yang selalu memiliki jalan keluar. Dalam algoritma *Recursive Division* memiliki 2 variabel input yang digunakan. Penjelasannya adalah sebagai berikut:

$$divide(0, x, 0, y) = \begin{cases} divide(0, x/2, 0, y) & \text{untuk } x = y \text{ rand} > 50 \\ divide(0, x, 0, y/2) & \text{untuk } x = y \text{ rand} < 50 \\ divide(0, x/2, 0, y) & \text{untuk } x > y \\ divide(0, x, 0, y/2) & \text{untuk } x < y \\ 0 & \text{untuk } x < 1 \text{ dan } y < 1 \end{cases}$$

Dari penjelasan tersebut diketahui bahwa  $x$  adalah nilai panjang labirin dan  $y$  adalah nilai lebar labirin. Sehingga akan banyak kondisi yang digunakan untuk pembangkitan *maze*. Untuk penjeleasan lebih detailnya adalah sebagai berikut:

1. Apabila panjang *maze* dan lebar *maze* bernilai sama, dan nilai random lebih dari 50 maka *maze* akan di *divide vertical*.
2. Apabila panjang *maze* dan lebar *maze* bernilai sama, dan nilai random lebih kecil dari 50 maka *maze* akan di *divide horizontal*.
3. Apabila panjang *maze* lebih besar dari lebar *maze* maka *maze* akan di *divide vertical*.
4. Apabila panjang *maze* lebih kecil dari lebar *maze* maka *maze* akan di *divide horizontal*.
5. Apabila nilai panjang sama dengan 1 dan lebar sama dengan 1, pembagian ruang berhenti.

### 3.2.3. Perancangan Level

Game AVIAR memiliki 16 level. Setiap level memiliki luas dimensi yang berbeda-beda. Karena setiap level memiliki luas dimensi yang berbeda maka jumlah bantuan yang akan diberikan juga berbeda-beda, selain itu lama waktu pergantian siang-malamnya juga akan berbeda-beda. Berikut pembagian rancangan setiap levelnya:

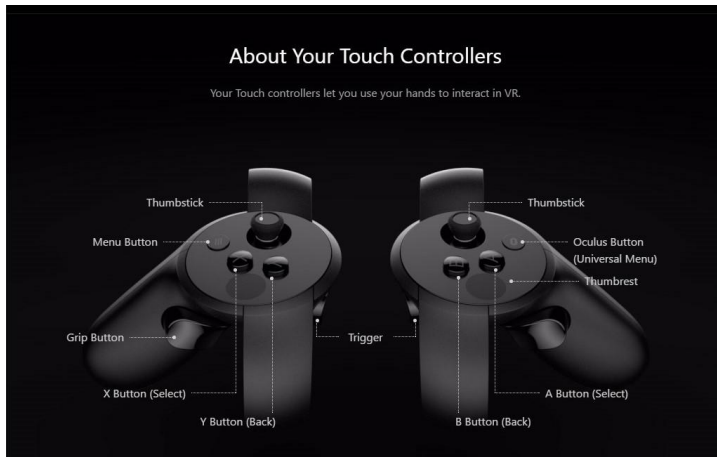
**Tabel 3.1 Konfigurasi *Level Game***

No	Level	Jumlah PowerUp	Dimensi
1	1	1	4x4
2	2	1	4x5
3	3	2	5x5
4	4	2	5x6
5	5	2	6x6
6	6	3	6x7
7	7	3	7x7
8	8	4	7x8
9	9	4	8x8
10	10	4	8x9
11	11	5	9x9
12	12	5	9x10
13	13	5	10x10
14	14	5	10x11
15	15	5	11x11
16	endless	9	12x12

### **3.2.4. Perancangan Pergerakan Karakter Pemain**

Permainan AVIAR ini menggunakan oculus rift sebagai *headmount* dan menggunakan oculus touch sebagai kontrolernya. Fungsi utama kontroler ini adalah untuk memilih, menekan, serta melakukan *grab* objek-objek yang ada didalam *game* dan menjalankan karakter. Tombol-tombol oculus touch dapat dilihat pada Gambar 3.5. Fungsi tombol oculus touch pada permainan bisa dilihat pada tabel 3.2





**Gambar 3.5 Tombol-tombol pada Oculus Touch**

(Sumber: <https://www.roadtovr.com/oculus-touch-review-reach-rift/>)

**Tabel 3.2 Tombol dan Fungsi Oculus Touch pada AVIAR**

Tombol	Fungsi
Left Thumbstick	Jalan karakter pemain
Right Thumstick	Rotasi karakter pemain
Trigger	Melakukan klik pada objek
Grip Button	Memegang sesuatu
Menu Button	Pause game saat bermain
X button	Kembali ke main menu

### 3.2.5. Perancangan Timer dalam Permainan

Permainan AVIAR memiliki *timer* yang berfungsi sebagai penanda waktu di dalam permainan yaitu sebagai pergantian siang dan malam dimana siang dan malam adalah waktu dimana labirin akan berubah bentuk dan musuh akan muncul. Namun didalam permainan AVIAR tidak memiliki batas waktu untuk kalah, jadi pemain bisa berputar-putar di dalam permainan sepuasnya.

### 3.2.6. Perancangan *Health* Pemain

Pemain dalam permainan AVIAR memiliki *health* sebagai ganti permainan tidak akan berakhir karena batas waktu seperti yang dijelaskan sebelumnya. *Health* pemain akan berkurang apabila pemain terkena serangan oleh musuh. *Health* pemain akan pulih apabila pemain mendapatkan *powerup* yang ada dalam permainan.

### 3.2.7. Perancangan *PowerUp*

*PowerUp* didalam permainan AVIAR merupakan objek bantuan untuk pemain. Seperti yang dijelaskan sebelumnya ada banyak macam *powerup* dalam permainan ini, yaitu, *speed up*, *health up*, *damage reduction*, *attack up*.

*Speed up* disini menjadikan kecepatan pemain bertambah tiga kali lipat. *Health up* berguna untuk menyembuhkan darah pemain sebanyak 50 *point*. *Damage reduction* menjadikan serangan musuh yang masuk ke pemain berkurang 50%. *Attack up* berguna untuk semua serangan pemain ke musuh meningkat 2 kali lipat.

### 3.2.8. Perancangan *Multiplayer*

Permainan AVIAR dapat dimainkan lebih dari 1 orang, agar permainan menjadi lebih seru dan lebih menarik. Jadi pemain bisa saling membantu satu sama lain saat memainkan *mingame* dan dalam mencari finish didalam permainan.

Pada permainan AVIAR *multiplayer* dijalankan cara pemain 1 masuk ke *lobby* dan menjadi *server*. Kemudian pemain ke 2 memasukkan *ip server* dan bergabung ke *lobby*. Setelah itu pemain 1 dan pemain 2 bisa memulai game bersama-sama.

Didalam labirin, pemain bisa berinteraksi satu sama lain di dunia nyata dengan langsung berbicara. Ketika salah satu pemain dikalahkan musuh, pemain lain masih bisa melanjutkan permainan. Ketika salah satu pemain menemukan finish, kedua pemain akan menang.

## 3.3. Perancangan Antarmuka

Pada subbab ini akan membahas bagaimana rancangan antarmuka pengguna yang akan dibangun pada permainan

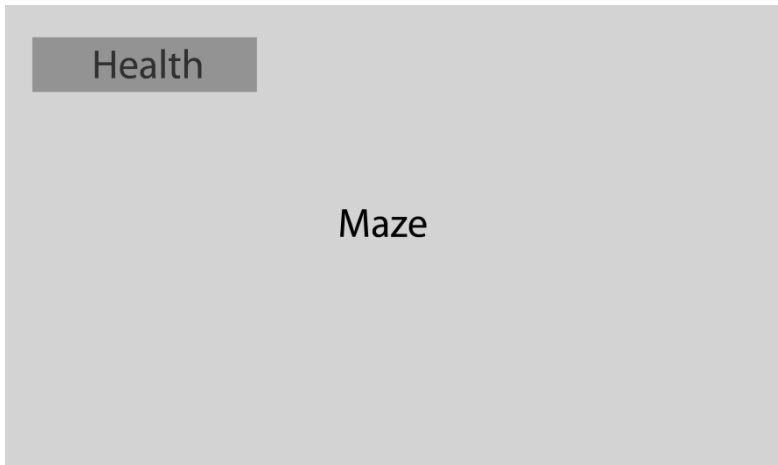
AVIAR. Permainan ini dibangun dengan mode *First Person Camera*(FPS). Rancangan antarmuka permainan bisa di lihat pada Gambar 3.6, Gambar 3.7, dan Gambar 3.8.



**Gambar 3.6 Rancangan Main Menu**



**Gambar 3.7 Rancangan Antarmuka Level Selection**



**Gambar 3.8 Rancangan Antarmuka Gameplay**

## **BAB IV**

### **IMPLEMENTASI SISTEM**

Bab ini membahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Implementasi yang dijelaskan adalah bagaimana menerapkan algoritma Recursive Division dalam game AVIAR untuk menghasilkan *random maze*. Game Engine yang digunakan adalah Unity dengan bahasa pemrograman C#.

#### **4.1. Lingkungan Pengembangan Sistem**

Lingkungan pengembangan sistem yang digunakan untuk mengembangkan Tugas Akhir ini dilakukan pada lingkungan pengembangan sebagai berikut:

**Tabel 4.1 Spesifikasi Lingkungan Implementasi**

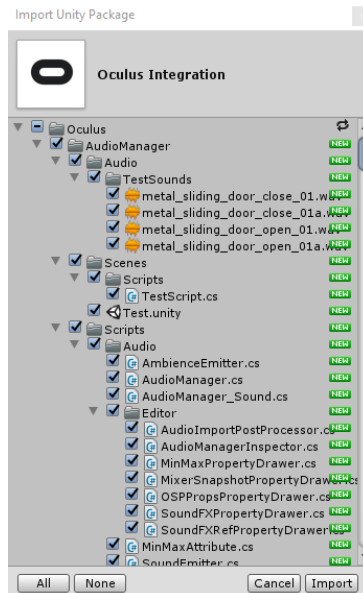
<b>Perangkat</b>	<b>Spesifikasi</b>
Perangkat Keras	<ul style="list-style-type: none"><li>• Prosesor Intel(R) Core (TM) i7-770 CPU @ 3.60GHz (8 CPUs), ~3.6 GHz</li><li>• RAM 8 Gigabyte</li><li>• Graphic card NVIDIA GeForce GTX 1060 3GB</li><li>• Oculus Rift</li></ul>
Perangkat Lunak	<ul style="list-style-type: none"><li>• Sistem operasi Windows 10 Home Single Language 64-bit</li><li>• Unity 2017.3.0f3 Personal (64-bit)</li></ul>

#### **4.2. Implementasi Pembuatan Aplikasi**

Dalam implementasi pembuatan aplikasi akan dibagi menjadi beberapa subbab yang dimulai dari jalannya *maze generator*, pembentukan maze, memilih stage, jalannya pemain, *control* pemain terhadap lingkungan sekitar, *timer* pergantian siang-malam. Implementasi dari semua fungsi diwujudkan dalam bentuk code dengan Bahasa pemrograman C#.

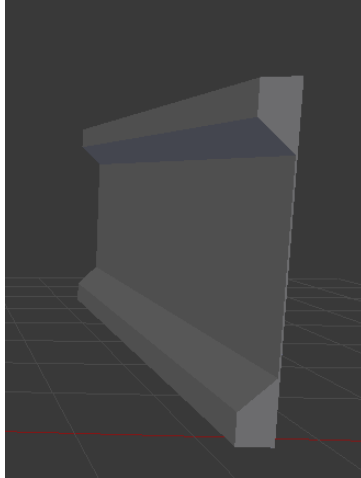
#### 4.2.1. Implementasi Import Asset dan Package

Permainan AVIAR ini berbasis *virtual reality* dan menggunakan perangkat keras oculus rift, jadi penulis menggunakan package oculus rift yang sudah *compatible* dengan unity. Ada banyak package yang disediakan unity untuk oculus rift yang bisa digunakan dan penulis menggunakan *package* oculus yang bernama “Oculus Integration” yang di unduh melalui *asset store* dari unity langsung secara gratis. Caranya dengan memilih *tab window* pada unity kemudian pilih *asset store*. Dalam *asset store* masukan kata kunci di kolom *search* yaitu oculus dan pilih oculus integration klik *download*, tunggu sampai selesai *mendownload* kemudian klik *import*. Kemudian akan keluar *window* seperti Gambar 4.1 klik *import* tunggu hingga proses *load asset* selesai dan *asset* berhasil diimport kedalam *project*.

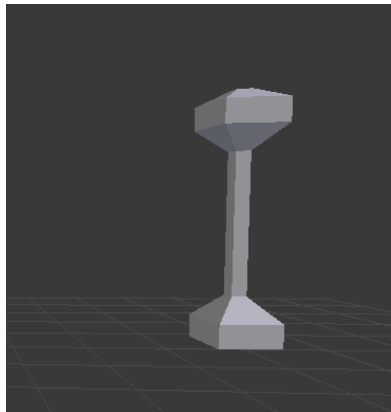


**Gambar 4.1 Implementasi Import Asset**

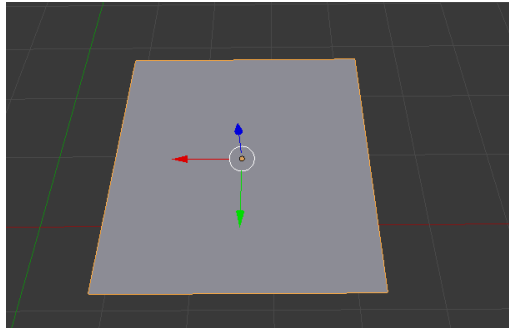
Pada pembuatan model labirin, ada beberapa objek yang harus dibuat yaitu lantai, dinding, pilar, *finish*, dan *powerup*. Objek yang dibuat oleh penulis sendiri adalah dinding seperti Gambar 4.2 , pilar pada Gambar 4.3, lantai yang bisa dilihat pada Gambar 4.4 dan *powerup* tipe *health* pada Gambar 4.5.



**Gambar 4.2 Objek Dinding**



**Gambar 4.3 Objek Pilar**



**Gambar 4.4 Objek Lantai**



**Gambar 4.5 Objek *Powerup Health***

Untuk *powerup* dan *finish* didapatkan penulis melalui *asset store* diantaranya adalah *double damage* pada Gambar 4.6, *damage reduction* pada Gambar 4.7, *speedup* pada Gambar 4.8, dan yang terakhir Gambar 4.9 adalah *finish*.





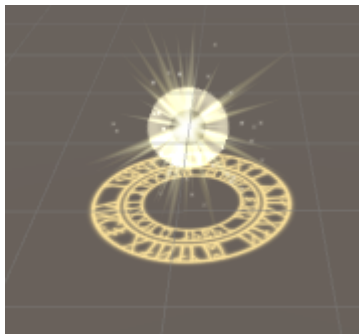
**Gambar 4.6 Objek *Powerup Double Damage***



**Gambar 4.7 Objek *Powerup Damage Reduction***



**Gambar 4.8 Objek *Powerup Speedup***



**Gambar 4.9 Objek *Finish***

#### **4.2.2. Implementasi Maze Generator**

Dalam pembuatan *maze generator* maka dapat dilihat pada Kode Sumber 4.1. Kode sumber tersebut menjelaskan tentang bagaimana dinding *maze terbentuk*. Langkah pertama yaitu membentuk dinding sisi luar sesuai ukuran *maze* yang diinginkan untuk menjadi dinding tepi *maze*. Kemudian menentukan dinding mana yang akan dihilangkan secara horisontal maupun vertikal. Pada proses ini dinding yang akan

dimunculkan diberi nilai *true*, sedangkan dinding yang diberi lubang akan diberi nilai *false*.

```

1.  public override void GenerateMaze(){
2.      for (int row = 0; row < RowCount; row++) {
3.          GetMazeCell(row,0).WallLeft = true;
4.          GetMazeCell(row,ColumnCount-1).WallRight= true;
5.      }
6.      for (int column = 0; column < ColumnCount; column++) {
7.          GetMazeCell(0,column).WallBack = true;
8.          GetMazeCell(RowCount-1,column).WallFront = true;
9.      }
10.
11.     rectsToDivide.Enqueue (new IntRect () {left = 0, right = ColumnCount, b
        ottom = 0, top = RowCount});
12.     while (rectsToDivide.Count > 0) {
13.         IntRect currentRect = rectsToDivide.Dequeue();
14.         int width = currentRect.right - currentRect.left;
15.         int height = currentRect.top - currentRect.bottom;
16.         if(width > 1 && height > 1){
17.             if(width > height){
18.                 divideVertical(currentRect);
19.             }else if(height > width){
20.                 divideHorizontal(currentRect);
21.             }else if(height == width){
22.                 if(Random.Range(0,100) > 50){
23.                     divideVertical(currentRect);
24.                 } else{
25.                     divideHorizontal(currentRect);
26.                 }
27.             }
28.             }else if(width > 1 && height <=1){
29.                 divideVertical(currentRect);
30.             }else if(width <=1 && height > 1){
31.                 divideHorizontal(currentRect);
32.             }
33.         }
34.     }
35.
36.     private void divideVertical(IntRect rect){
37.         int divCol = Random.Range(rect.left, rect.right-1);
38.         for (int row = rect.bottom; row < rect.top; row++) {

```

```

39.         GetMazeCell(row,divCol).WallRight = true;
40.         GetMazeCell(row,divCol+1).WallLeft = true;
41.     }
42.     int space = Random.Range (rect.bottom, rect.top);
43.     GetMazeCell (space, divCol).WallRight = false;
44.     if (divCol + 1 < rect.right) {
45.         GetMazeCell (space, divCol + 1).WallLeft = false;
46.     }
47.     rectsToDivide.Enqueue (new IntRect () {left = rect.left, right = divCol+1,
        bottom = rect.bottom, top = rect.top});
48.     rectsToDivide.Enqueue (new IntRect () {left = divCol+1, right = rect.right
        , bottom = rect.bottom, top = rect.top});
49. }
50.
51. private void divideHorizontal(IntRect rect){
52.     int divRow = Random.Range(rect.bottom, rect.top-1);
53.     for (int col = rect.left; col < rect.right; col++) {
54.         GetMazeCell(divRow,col).WallFront = true;
55.         GetMazeCell(divRow+1,col).WallBack = true;
56.     }
57.     int space = Random.Range (rect.left, rect.right);
58.     GetMazeCell (divRow, space).WallFront = false;
59.     if (divRow + 1 < rect.top) {
60.         GetMazeCell (divRow + 1, space).WallBack = false;
61.     }
62.     rectsToDivide.Enqueue (new IntRect () {left = rect.left, right = rect.right,
        bottom = rect.bottom, top = divRow+1});
63.     rectsToDivide.Enqueue (new IntRect () {left = rect.left, right = rect.right,
        bottom = divRow+1, top = rect.top});
64. }

```

### Kode Sumber 4.1 Implementasi *Maze Generator*

#### 4.2.3. Implementasi *Level*

Pada pembuatan *level* karena permainan ini memiliki banyak *level*, maka dibutuhkan banyak scene juga sebagai penempatan *maze* tiap levelnya. Implementasinya dengan cara membuat *GameObject* kosong pada scene kemudian menambahkan *source code* yang bisa dilihat pada kode sumber, kemudian tinggal memasukkan *row* dan *coloumn* sesuai yang diinginkan.

```

1. using UnityEngine;
2. using System.Collections;
3. using System.Collections.Generic;
4. using UnityEngine.Networking;
5.
6. public int Rows;
7. public int Columns;
8.
9. public class MazeSpawner : NetworkBehaviour
10. {
11.     public void Mazemaker()
12.     {
13.         mMazeGenerator = new DivisionMazeGenerator(Rows, Column
ns);
14.     }
15. }

```

### Kode Sumber 4.2 Implementasi *Level*

#### 4.2.4. Implementasi Pergerakan Karakter Pemain

Pada saat bermain, pemain dapat menggerakkan karakter pemain dengan menggunakan *controller* oculus. Pemain bisa bergerak bebas, melakukan touch, dan grab. Untuk *movement* pemain bisa di liat di Kode Sumber 4.2.

```

1. public virtual void UpdateMovement()
2. {
3.
4.     bool moveForward = Input.GetKey(KeyCode.W) || Input.GetKey(KeyCode.UpArrow);
5.     bool moveLeft = Input.GetKey(KeyCode.A) || Input.GetKey(KeyCode.LeftArrow);
6.     bool moveRight = Input.GetKey(KeyCode.D) || Input.GetKey(KeyCode.RightArrow);
7.     bool moveBack = Input.GetKey(KeyCode.S) || Input.GetKey(KeyCode.DownArrow);
8.
9.
10.     bool dpad_move = false;

```

```

11.
12.     if (OVRInput.Get(OVRInput.Button.DpadUp))
13.     {
14.         moveForward = true;
15.         dpad_move = true;
16.
17.     }
18.
19.     if (OVRInput.Get(OVRInput.Button.DpadDown))
20.     {
21.         moveBack = true;
22.         dpad_move = true;
23.     }
24.
25.
26.     if (SnapRotation)
27.     {
28.
29.         if (OVRInput.Get(OVRInput.Button.SecondaryThumbstickLeft))
30.         {
31.             if (ReadyToSnapTurn)
32.             {
33.                 euler.y -= RotationRatchet;
34.                 ReadyToSnapTurn = false;
35.             }
36.         }
37.         else if (OVRInput.Get(OVRInput.Button.SecondaryThumbstickRight))
38.         {
39.             if (ReadyToSnapTurn)
40.             {
41.                 euler.y += RotationRatchet;
42.                 ReadyToSnapTurn = false;
43.             }
44.         }
45.         else
46.         {
47.             ReadyToSnapTurn = true;
48.         }
49.     }
50.     else
51.     {

```

```

52.     Vector2 secondaryAxis = OVRInput.Get(OVRInput.Axis2D.S
    econdaryThumbstick);
53.     euler.y += secondaryAxis.x * rotateInfluence;
54. }
55.
56.     transform.rotation = Quaternion.Euler(euler);
57. }
58. }

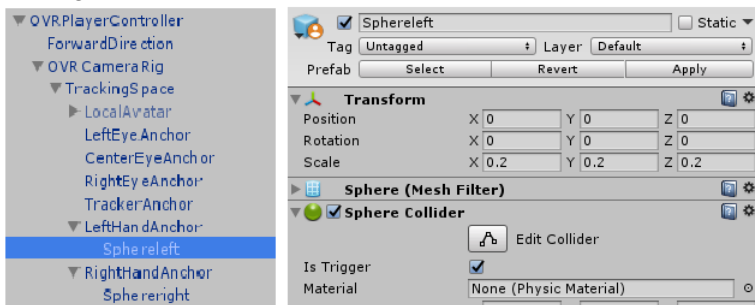
```

### Kode Sumber 4.3 Implementasi Gerakan Pemain

#### 4.2.5. Implementasi Interaksi dengan Objek

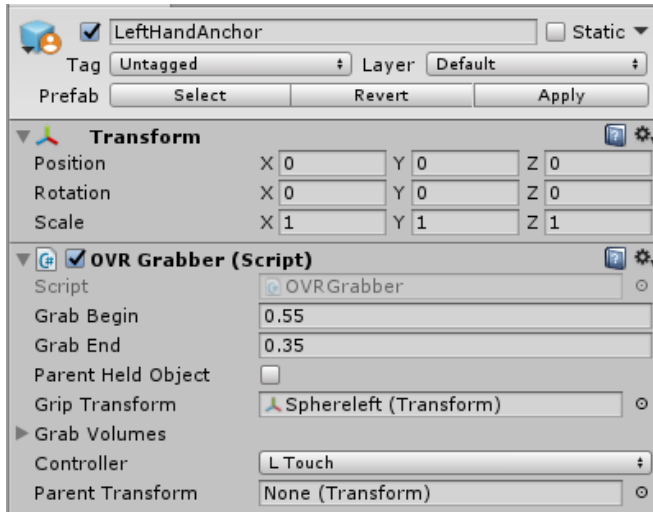
Pergerakan pemain selain *movement* yaitu ada grab. Untuk melakukan grab penulis menggunakan script default dari *package* oculus yang sudah diimport, langkah-langkahnya adalah sebagai berikut:

1. Buka prefab OVRPlayerController pada asset oculus.
2. Membuat sebuah *collider* didalam *handanchor*. Disini player menggunakan *sphere collider*. Bisa dilihat pada gambar 4.10.



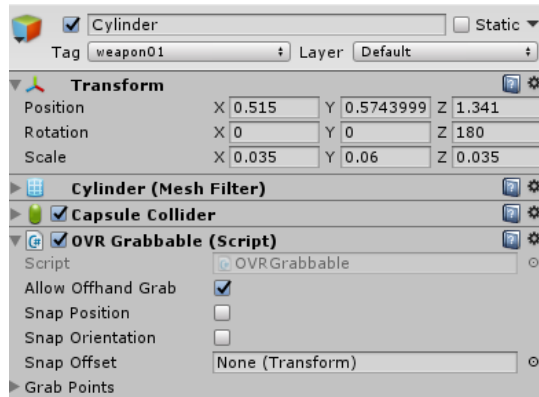
**Gambar 4.10 Sphere Collider pada prefab karakter**

3. Didalam *handanchor* tambahkan komponen berupa script OVR Grabber. Bisa dilihat di gambar 4.11.



**Gambar 4.11 Grabber pada handanchor**

4. Kemudian tambahkan OVR Grabbable ke objek-objek yang bisa digrab/digenggam oleh pemain, seperti gambar 4.12.



**Gambar 4.12 Grabbable pada objek**



#### 4.2.6. Implementasi *Timer* dalam Permainan

Dalam Permainan AVIAR terdapat *timer* yang mengontrol siang-malam, jadi disini pergantian siang-malam disini diharapkan pemain bisa merasakan pengalaman bermain *game* dengan lingkungan senyata mungkin. Lama pergantian waktu siang-malam bergantung pada tiap level. Untuk membuat siang malam membutuhkan 2 *directional light* gelap terang yang saling berhadapan, kemudian tambahkan kode seperti kode sumber 4.3.

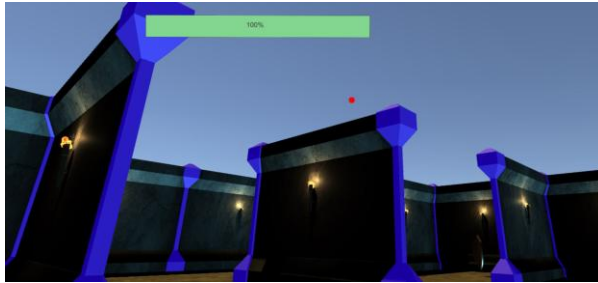
```

1.  using System.Collections;
2.  using System.Collections.Generic;
3.  using UnityEngine;
4.  using UnityEngine.Networking;
5.
6.  public class Sun : NetworkBehaviour {
7.      private OVRPlayerController ovrp;
8.
9.      void Start () {
10.         ovrp = FindObjectOfType<OVRPlayerController> ();
11.     }
12.
13.     void Update () {
14.         if (isServer) {
15.
16.             if (ovrp.setpause == false) {
17.                 transform.RotateAround (Vector3.zero, Vector3.right, 5.8f *
Time.deltaTime);
18.                 transform.LookAt (Vector3.zero);
19.             }
20.         }
21.     }
22. }

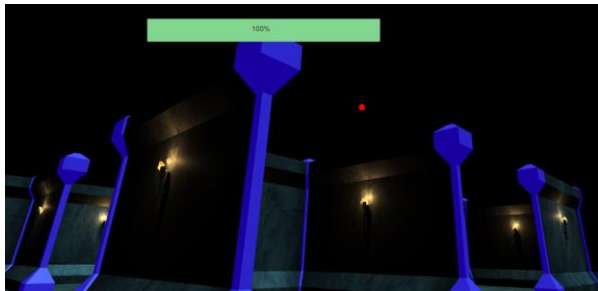
```

#### Kode Sumber 4.4 Implementasi Siang dan Malam

Hasil implementasi keadaan siang dan malam waktu berada didalam labirin dapat dilihat pada Gambar 4.13 dan Gambar 4.14.



**Gambar 4.13 Labirin Saat Siang**



**Gambar 4.14 Labirin Saat Malam**

Timer didalam *game* AVIAR juga digunakan sebagai kontrol eksekusi *random maze* dan kontrol musuh yang keluar pada waktu malam hari. Implementasi kodenya dapat dilihat pada kode sumber 4.4.

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.Networking;
5.
6. public class Timer : NetworkBehaviour
7. {
8.     public AudioSource audiomonspawn;
9.     public float timer;
10.    public bool waktu;
11.    public float waktucek;
12.    public bool spawned;

```

```

13.  public bool forsun;
14.  private MazeSpawner mazeSpawner;
15.  private OVRPlayerController ovrp;
16.  private Finished finishhh;
17.  private Phealth healthp;
18.
19.  public List<Transform> spawnPoint = new List<Transform>();
20.  public GameObject monsterPrefab;
21.
22.  void Awake()
23.  {
24.      mazeSpawner = gameObject.GetComponent<MazeSpawner>();
25.  }
26.  // Use this for initialization
27.  void Start()
28.  {
29.      ovrp = FindObjectOfType<OVRPlayerController> ();
30.      finishhh = FindObjectOfType<Finished>().GetComponent<Finis
hed>();
31.      timer = 0;
32.      healthp = FindObjectOfType<Phealth> ().GetComponent<Phealt
h> ();
33.      forsun = false;
34.
35.      waktu = false;
36.  }
37.
38.  // Update is called once per frame
39.  void Update()
40.  {
41.      if (ovrp.setpause == true) {
42.          forsun = true;
43.          return;
44.      }
45.
46.      if ( finishhh.setfinish == true || healthp.over == true) {
47.          mazeSpawner.DesMons();
48.          forsun = true;
49.          return;
50.      }
51.
52.      forsun = false;
53.

```

```

54.     if (isServer) {
55.         Check ();
56.     }
57. }
58.
59. private void SpawnMonster(int spawnPointIndex) {
60.     GameObject temp = Instantiate(monsterPrefab, spawnPoint[spawnPointIndex].position, spawnPoint[spawnPointIndex].rotation);
61.     NetworkServer.Spawn(temp);
62.
63. }
64.
65. private void Check()
66. {
67.
68.
69.     if (timer % waktucek < 1 && waktu == false && spawned == false )
70.     {
71.
72.         waktu = true;
73.         spawned = true;
74.
75.         mazeSpawner.DesMons();
76.         mazeSpawner.Mazemaker ();
77.         cam3rd.skycount += 1;
78.     }
79.
80.     else if (timer % waktucek < 1 && waktu == true && spawned == false)
81.     {
82.         for (int i = 0; i < spawnPoint.Count; i++) {
83.             SpawnMonster(i);
84.         }
85.         audiomonsspawn.Play ();
86.         waktu = false;
87.         spawned = true;
88.         mazeSpawner.Mazemaker ();
89.         cam3rd.skycount += 1;
90.
91.     }
92.
93.     else if (timer % waktucek > 1 && spawned == true)

```

```

94.     {
95.         spawned = false;
96.     }
97.
98.     }
99. }
100.

```

#### Kode Sumber 4.5 Implementasi *Timer Random Maze*

##### 4.2.7. Implementasi *Health* pemain

Permainan AVIAR ini menggunakan *health* sebagai kontrol pemain apabila pemain terserang oleh musuh. Tujuannya agar pemain lebih berhati-hati dan *health* juga bisa bertambah apabila pemain menemukan *power up* tipe *healthup*. Kode Sumber 4.5 berikut adalah implementasi dari *health*.

```

1.  using System.Collections;
2.  using System.Collections.Generic;
3.  using UnityEngine;
4.  using UnityEngine.UI;
5.
6.  public class HealthBar : MonoBehaviour {
7.
8.      private Phealth healthp;
9.      public Image currentHealthBar;
10.     public Text ratioText;
11.     private float ratio;
12.
13.     // Use this for initialization
14.     void Start () {
15.         healthp = gameObject.GetComponent<Phealth>();
16.         UpdateHealthBar();
17.         healthp.hp = 100;
18.         healthp.maxhp = 100;
19.     }
20.
21.     // Update is called once per frame
22.     public void UpdateHealthBar () {
23.         ratio = healthp.hp / healthp.maxhp;
24.         currentHealthBar.rectTransform.localScale = new Vector3(ratio,
1, 1);

```

```

25.         ratioText.text = (ratio * 100).ToString() + '%';
26.     }
27. }

```

### Kode Sumber 4.6 Implementasi *Health* Pemain

#### 4.2.8. Implementasi *PowerUp*

Pada implementasi *powerup game* AVIAR ini bertujuan sebagai bahan bantuan untuk pemain, bisa berupa buff bisa berupa *restoration*. *Powerup* di *game* ini banyak macamnya, yaitu *speed up*, *health up*, *damage reduction*, *attack up*. Untuk mendapatkannya bisa dibilang cukup mudah karena mereka ada didalam *maze* namun posisinya random dan tanpa harus melakukan sesuatu yang spesial seperti menjawab soal. Jadi, pemain tinggal jalan-jalan berkeliling labirin apabila pemain beruntung pemain bisa mendapatkannya. Implementasi *powerup* ini bisa dilihat pada Kode Sumber 4.6 sebagai berikut.

```

1.  using System.Collections;
2.  using System.Collections.Generic;
3.  using UnityEngine;
4.  using UnityEngine.SceneManagement;
5.  using UnityEngine.Networking;
6.
7.  public class Phealth : NetworkBehaviour {
8.
9.      public AudioSource powup;
10.     public AudioSource lose;
11.     public bool ceklose;
12.     public float hp = 100 ;
13.     public float maxhp = 100 ;
14.     private float y;
15.     public float cd;
16.     public bool dmgred;
17.     public float dmred;
18.     public bool spd;
19.     public bool dualdmg;
20.     public float dbldmg;
21.     private float countdown;
22.     private float countdown1;
23.     public float overcount;

```

```

24. public bool over;
25. private HealthBar barhealth;
26. private GameObject fail;
27.
28. void Start () {
29.     barhealth = gameObject.GetComponent<HealthBar>();
30.     spd = false;
31.     dmgred = false;
32.     dualdmg = false;
33.     cd = 0;
34.     dmred = 1;
35.     dbldmg = 1;
36.     overcount = 0;
37.
38.     over = false;
39.     ceklose = false;
40. }
41.
42. // Update is called once per frame
43. void Update () {
44.     gameOver();
45.     audiogameover ();
46.     if(dmgred == true)
47.     {
48.         countdown = countdown - Time.deltaTime * 1.0f;
49.         Debug.Log(dmgred);
50.         dmred = 2;
51.         if (countdown <= 0)
52.         {
53.             dmgred = false;
54.
55.         }
56.     }
57.     if(dualdmg == true)
58.     {
59.         countdown1 = countdown1 - Time.deltaTime * 1.0f;
60.         Debug.Log(dualdmg);
61.         dbldmg = 2;
62.         if (countdown1 <= 0)
63.         {
64.             dualdmg = false;
65.             dbldmg = 2;
66.         }

```

```

67.     }
68. }
69.
70. private void gameOver()
71. {
72.     if (hp <= 0)
73.     {
74.         over = true;
75.     }
76. }
77. private void audiogameover(){
78.     if (over == true)
79.     {
80.         overcount += 1;
81.     }
82.     if (overcount == 1)
83.     {
84.
85.         lose.Play();
86.     }
87. }
88.
89. void OnTriggerEnter (Collider other){
90.     if (other.gameObject.tag == "hpup") {
91.         powup.Play ();
92.         this.hp += 50;
93.         if (hp > 100) {
94.             hp = 100;
95.         }
96.         Destroy (other.gameObject);
97.         Debug.Log (hp);
98.         barhealth.UpdateHealthBar();
99.     }
100.
101.     if ( other.gameObject.tag == "Deff") {
102.         powup.Play ();
103.         dmgrid = true;
104.         countdown = 15;
105.         Destroy(other.gameObject);
106.     }
107.
108.     if ( other.gameObject.tag == "DD") {
109.         powup.Play ();

```



```

110.     dualdmg = true;
111.     countdown1 = 15;
112.     Destroy(other.gameObject);
113. }
114.
115. if (other.gameObject.tag == "testing")
116. {
117.     powup.Play ();
118.     spd = true;
119.     cd = 10;
120.     Destroy(other.gameObject);
121. }
122.
123. if (other.gameObject.tag == "Monster")
124. {
125.     if (dmgred == false)
126.     {
127.         this.hp -= 2 / dmred;
128.         Debug.Log(hp);
129.         if (hp < 0)
130.         {
131.             hp = 0;
132.             ceklose = true;
133.             barhealth.UpdateHealthBar();
134.         }
135.         barhealth.UpdateHealthBar();
136.
137.     }
138.
139.     else if (dmgred == true && countdown >= 0)
140.     {
141.         this.hp -= 2 / dmred;
142.         if (hp < 0)
143.         {
144.             hp = 0;
145.             ceklose = true;
146.             barhealth.UpdateHealthBar();
147.         }
148.         Debug.Log(hp);
149.         barhealth.UpdateHealthBar();
150.     }
151.     barhealth.UpdateHealthBar();
152. }

```

```

153.
154. }
155.
156. [Command]
157. void CmdFail() {
158.
159.     RpcFail();
160.     //Debug.Log(gameObject.name);
161.     if (over == true)
162.     {
163.         overcount += 1;
164.     }
165.     if (overcount == 1)
166.     {
167.
168.         lose.Play();
169.     }
170. }
171.
172.
173. [ClientRpc]
174. void RpcFail() {
175.     Debug.Log(gameObject.name);
176.     if (over == true)
177.     {
178.         overcount += 1;
179.     }
180.     if (overcount == 1)
181.     {
182.
183.         lose.Play();
184.     }
185. }
186.
187.
188. }
189.

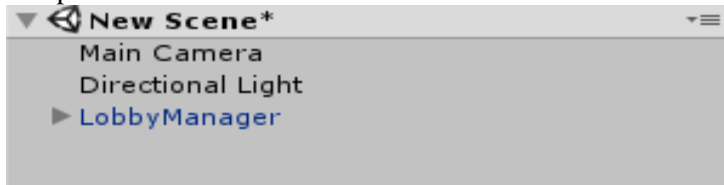
```

**Kode Sumber 4.7 Implementasi *PowerUp***

#### 4.2.9. Implementasi *Multiplayer*

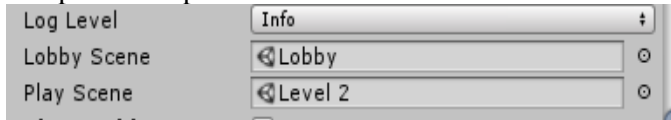
Dalam Implementasi *multiplayer* penulis menggunakan aset dari unity *asset store*, yaitu menggunakan Network Lobby Manager. Tahapan pembuatannya yaitu:

1. Membuat *scene* kosong.
2. Masukkan Lobby manager kedalam *scene*, dapat dilihat pada Gambar 4.15.



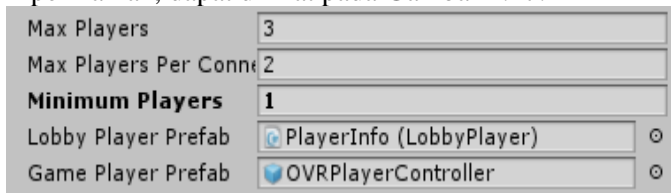
**Gambar 4.15 Memasukan *Network Lobby* ke *scene* baru**

3. Masukkan *lobby scene* dan *scene* yang akan dimainkan, dapat dilihat pada Gambar 4.16.



**Gambar 4.16 Memasukkan *lobby scene* dan *play scene***

4. Setelah itu masukkan karakter yang akan digunakan dalam permainan, dapat dilihat pada Gambar 4.17.



**Gambar 4.17 Memasukkan objek karakter**

5. Terakhir masukkan objek-objek yang akan dimunculkan di dalam permainan, dapat dilihat pada Gambar 4.18.

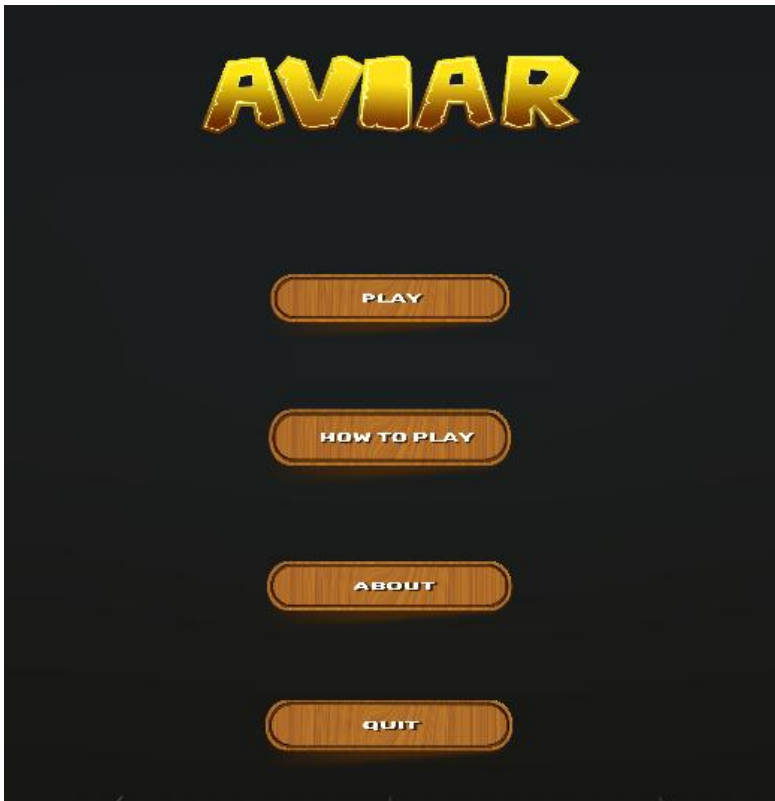


**Gambar 4.18 Memasukkan *spawnable* objek**

### 4.3. Implementasi Antarmuka

#### 4.3.1. Implementasi Antarmuka *Main Menu*

Implementasi antarmuka *Main Menu* mencakup tombol menuju ke *stage selection* yaitu dengan menggunakan tombol 'play', tombol 'how to play' untuk melihat penjelasan tentang *game*, 'about' berisi tentang data penulis dan tombol 'quit' untuk keluar aplikasi. Hasil implementasi *main menu* dapat dilihat pada Gambar 4.19 dan kode sumber bisa dilihat pada Kode Sumber 4.8.



**Gambar 4.19 Implementasi Antarmuka *Main Menu***

```
1. public void OnPlayClick()
2. {
3.     mmButtons.SetActive(false);
4.     levelselect.SetActive(true);
5.     backButton.SetActive(true);
6.     posisiPadaMM = "lvs";
7. }
8.
9. public void OnHowToPlayClick()
10. {
11.     mmButtons.SetActive(false);
```

```

12. backButton.SetActive(true);
13. http1.SetActive(true);
14. posisiPadaMM = "http1";
15. }
16.
17. public void OnAboutClick()
18. {
19.     posisiPadaMM = "abt";
20.     mmButtons.SetActive(false);
21.     about.SetActive(true);
22.     backButton.SetActive(true);
23. }
24.
25. public void OnQuitClick()
26. {
27.     Application.Quit();
28.
29. }

```

#### Kode Sumber 4.8 Implementasi *Main Menu*

##### 4.3.2. Implementasi Antarmuka *Level Selection*

Implementasi antarmuka *Level Selection* mencakup daftar *stage* yang dapat diakses oleh pemain dan dapat dimainkan, serta pilihan untuk kembali ke main menu. Pengimplimentasian dapat dilihat pada Gambar 4.20 dan Kode Sumber 4.9



**Gambar 4.20 Implementasi Antarmuka *Level Selection***

```

1.  public void OnLevel1Select()
2.  {
3.      SceneManager.LoadScene("Lobby");
4.      PlayerPrefs.SetString("Level", "1");
5.  }
6.
7.  public void OnLevel2Select()
8.  {
9.      SceneManager.LoadScene("Lobby");
10.     PlayerPrefs.SetString("Level", "2");
11. }
12.
13. public void OnLevel3Select()
14. {
15.     SceneManager.LoadScene("Lobby");
16.     PlayerPrefs.SetString("Level", "3");
17. }
18.

```

```
19. public void OnLevel4Select()
20. {
21.     SceneManager.LoadScene("Lobby");
22.     PlayerPrefs.SetString("Level", "4");
23. }
24.
25. public void OnLevel5Select()
26. {
27.     SceneManager.LoadScene("Lobby");
28.     PlayerPrefs.SetString("Level", "5");
29. }
30.
31. public void OnLevel6Select()
32. {
33.     SceneManager.LoadScene("Lobby");
34.     PlayerPrefs.SetString("Level", "6");
35. }
36.
37. public void OnLevel7Select()
38. {
39.     SceneManager.LoadScene("Lobby");
40.     PlayerPrefs.SetString("Level", "7");
41. }
42.
43. public void OnLevel8Select()
44. {
45.     SceneManager.LoadScene("Lobby");
46.     PlayerPrefs.SetString("Level", "8");
47. }
48.
49. public void OnLevel9Select()
50. {
51.     SceneManager.LoadScene("Lobby");
52.     PlayerPrefs.SetString("Level", "9");
53. }
54.
55. public void OnLevel10Select()
56. {
57.     SceneManager.LoadScene("Lobby");
58.     PlayerPrefs.SetString("Level", "10");
59. }
60.
61. public void OnLevel11Select()
```



```

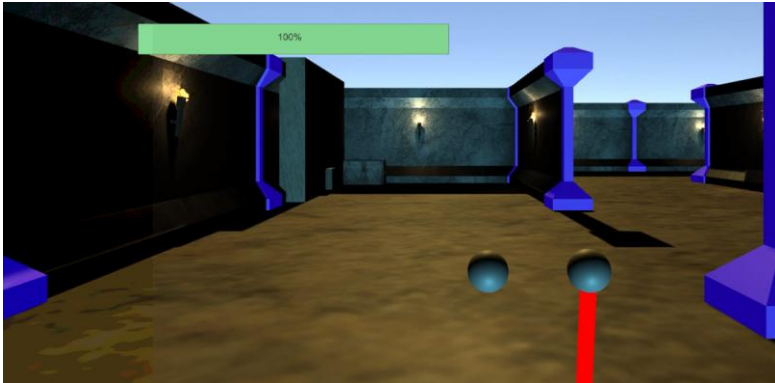
62. {
63.     SceneManager.LoadScene("Lobby");
64.     PlayerPrefs.SetString("Level", "11");
65. }
66.
67. public void OnLevel12Select()
68. {
69.     SceneManager.LoadScene("Lobby");
70.     PlayerPrefs.SetString("Level", "12");
71. }
72.
73. public void OnLevel13Select()
74. {
75.     SceneManager.LoadScene("Lobby");
76.     PlayerPrefs.SetString("Level", "13");
77. }
78.
79. public void OnLevel14Select()
80. {
81.     SceneManager.LoadScene("Lobby");
82.     PlayerPrefs.SetString("Level", "14");
83. }
84.
85. public void OnLevel15Select()
86. {
87.     SceneManager.LoadScene("Lobby");
88.     PlayerPrefs.SetString("Level", "15");
89. }
90.
91. public void OnLevelrallySelect()
92. {
93.     SceneManager.LoadScene("Lobby");
94.     PlayerPrefs.SetString("Level", "Rally");
95. }

```

**Kode Sumber 4.9 Implementasi Antarmuka *Level Selection***

#### 4.3.3. Implementasi Antarmuka *Gameplay Stage*

Implementasi antarmuka *Gameplay Stage* mencakup komponen *stage*. Pada bagian atas layar permainan terdapat *healthbar* pemain selain itu juga terdapat pause dan bisa mengarahkan pemain kembali ke *main menu* awal. Hasil implementasi bisa dilihat pada Gambar 4.21.



**Gambar 4.21 Implementasi Antarmuka *Gameplay Stage***

## **BAB V**

### **PENGUJIAN DAN EVALUASI**

#### **5.1. Lingkungan Pengujian**

Lingkungan pelaksanaan uji coba meliputi perangkat keras dan perangkat lunak yang akan digunakan pada sistem ini. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam rangka uji coba perangkat lunak ini dicantumkan pada Tabel 5.1.

**Tabel 5.1 Lingkungan Pengujian**

<b>Perangkat</b>	<b>Spesifikasi</b>
Perangkat Keras	<ul style="list-style-type: none"><li>• Prosesor Intel(R) Core (TM) i7-770 CPU @ 3.60GHz (8 CPUs), ~3.6 GHz</li><li>• RAM 8 Gigabyte</li><li>• Graphic card NVIDIA GeForce GTX 1060 3GB</li><li>• Oculus Rift</li></ul>
Perangkat Lunak	<ul style="list-style-type: none"><li>• Sistem operasi Windows 10 Home Single Language 64-bit</li><li>• Unity 2017.3.0f3 Personal (64-bit)</li></ul>

#### **5.2. Pengujian Fungsionalitas**

Pengujian Fungsionalitas ini untuk mengetahui kesesuaian keluaran dari tiap tahap dan langkah penggunaan fitur terhadap skenario yang disiapkan.

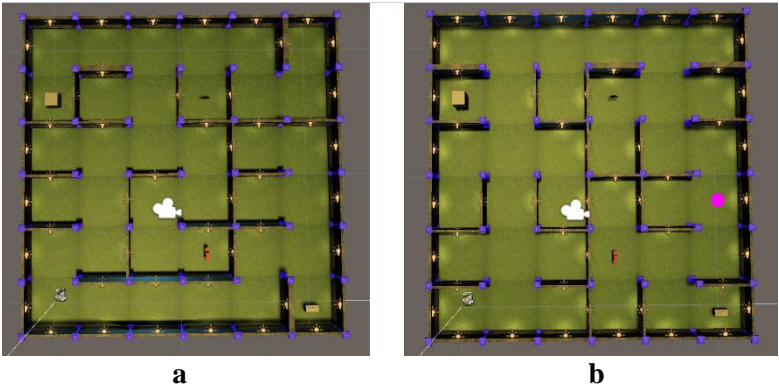
Pengujian ini dilakukan untuk menguji apakah fungsionalitas yang diidentifikasi benar-benar diimplementasikan dan bekerja sebagaimana seharusnya. Pengujian juga dilakukan untuk mengetahui kesesuaian setiap tahapan skenario. Skenario uji coba fungsionalitas yang dilakukan terhadap aplikasi yang dibangun dijelaskan pada Tabel 5.2.

**Tabel 5.2 Skenario Uji Coba Fungsionalitas**

Kode Uji Coba	Nama Uji Coba
UC-001	Uji Coba Pembangkit <i>Maze</i>
UC-002	Uji Coba Multiplayer
UC-003	Uji Coba Pergerakan Pemain
UC-004	Uji Coba Interaksi dengan Objek
UC-005	Uji Coba Menang Kalah
UC-006	Uji Coba <i>Main Menu</i>
UC-007	Uji Coba <i>Level Selection</i>

**5.2.1. Pengujian Pembangkit *Maze***

Pengujian Pembangkit *Maze* dilakukan dengan cara melihat hasil pembangkitan *maze* pada waktu siang hari dan malam hari. Hasil uji coba dapat dilihat pada Gambar 5.1 dan Tabel 5.3.



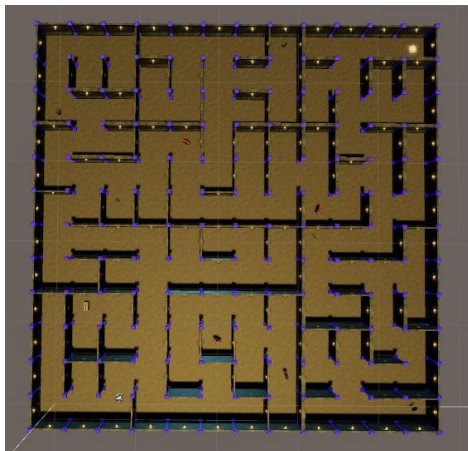
**Gambar 5.1 a.Bentuk *maze* level 5 pada saat siang hari  
b.Bentuk *maze* level 5 pada saat malam hari**

**Tabel 5.3 Hasil Uji Coba Pembangkit *Maze***

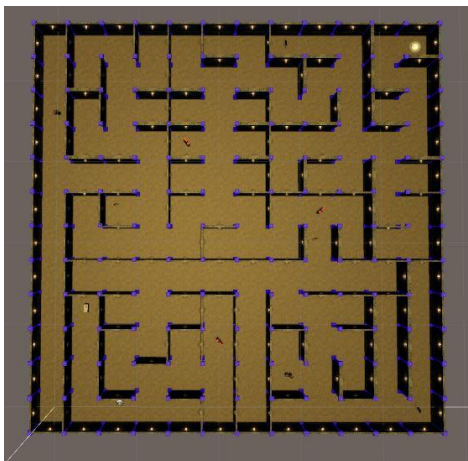
ID	UC-001
Nama	Uji Coba Pembangkit <i>Maze</i> .
Tujuan	Mengetahui hasil random <i>maze</i> .

Skenario 1. (Join game dari lobby)	
Kondisi awal	Pemain berada di <i>Network Lobby</i> .
Masukan	Fungsi tombol join pada <i>Network Lobby</i> .
Keluaran yang diharapkan	Labirin muncul sempurna pada <i>game</i> .
Kondisi akhir	Labirin muncul.
Skenario 2. (Menunggu reset waktu siang malam)	
Kondisi awal	Pemain berada di <i>maze</i> .
Masukan	Menunggu reset waktu siang dan malam.
Keluaran yang diharapkan	Labirin berubah bentuk dari yang sebelumnya.
Kondisi akhir	Labirin berubah bentuk.
Hasil Ujicoba	Berhasil

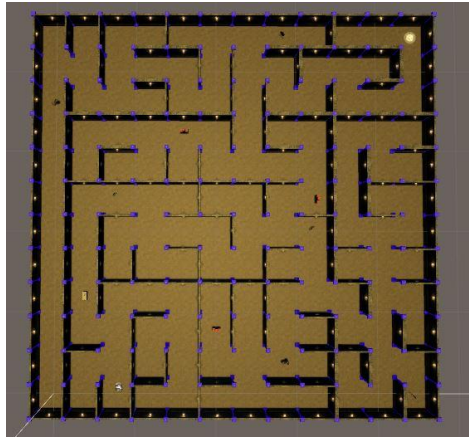
Pengujian dilakukan dengan cara mengulang level endless secara 5 kali berturut-turut sebagai bukti bahwa bentuk *maze* akan berubah setiap kali permainan dimulai. Hasil pengujian bisa dilihat pada Gambar 5.2, Gambar 5.3, Gambar 5.4, Gambar 5.5, dan Gambar 5.6.



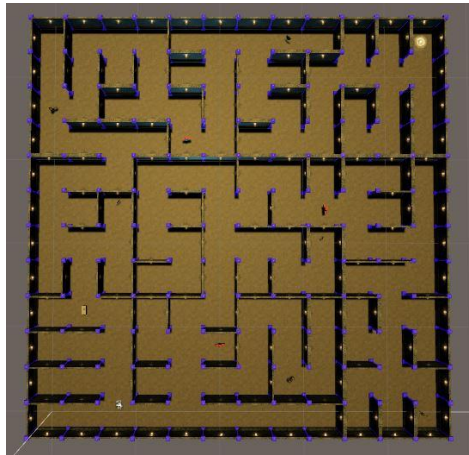
**Gambar 5.2 Bentuk *Maze* Percobaan 1**



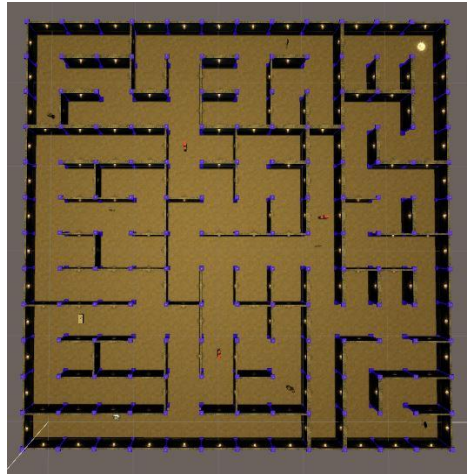
**Gambar 5.3 Bentuk *Maze* Percobaan 2**



**Gambar 5.4 Bentuk *Maze* Percobaan 3**



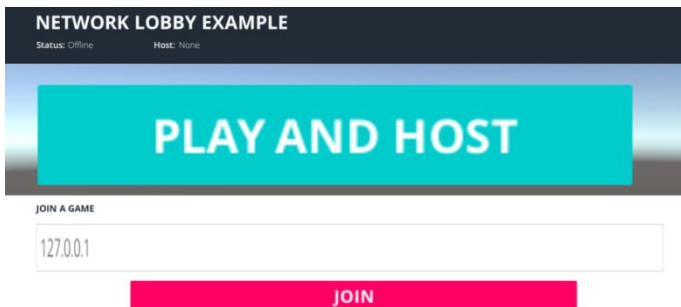
**Gambar 5.5 Bentuk *Maze* Percobaan 4**



**Gambar 5.6 Bentuk Maze Percobaan 5**

### **5.2.2. Pengujian Multiplayer**

Pada layar *Network Lobby* pemain bisa memilih apakah mau bermain sendirian atau multi pemain. Jika pemain ingin bermain seorang diri bisa memilih tombol ‘Play and Host’ seperti pada Gambar 5.7. dan kemudian langsung memilih join seperti Gambar 5.8.



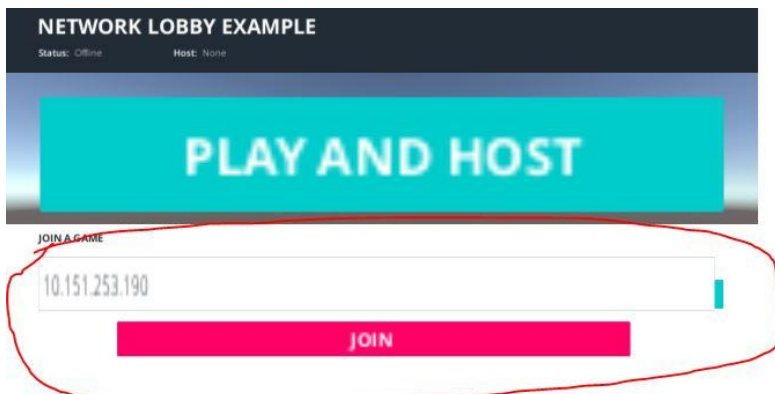
**Gambar 5.7 Tampilan Network Lobby**



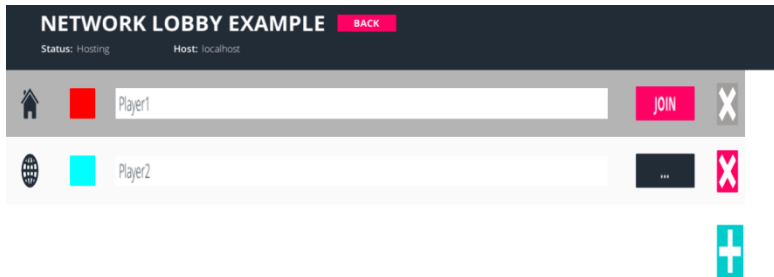


**Gambar 5.8 Tampilan *Network Lobby***

Apabila pemain ingin bermain multi pemain, maka salah satu pemain harus menjadi *server* dan yang lainnya menjadi *client* caranya adalah si *server* memilih tombol ‘Play and Host’ kemudian menunggu si *client*. Agar si *client* bisa bergabung *client* harus memasukkan *IP address server* kemudian pilih tombol ‘Join’ seperti contoh pada Gambar 5.9. Setelah *client* bergabung dengan *server* kedua pemain klik tombol ‘Join’ seperti gambar 5.10. Hasil uji coba bisa dilihat pada Tabel 5.4.



**Gambar 5.9 Tampilan *input IP* pada *Client***



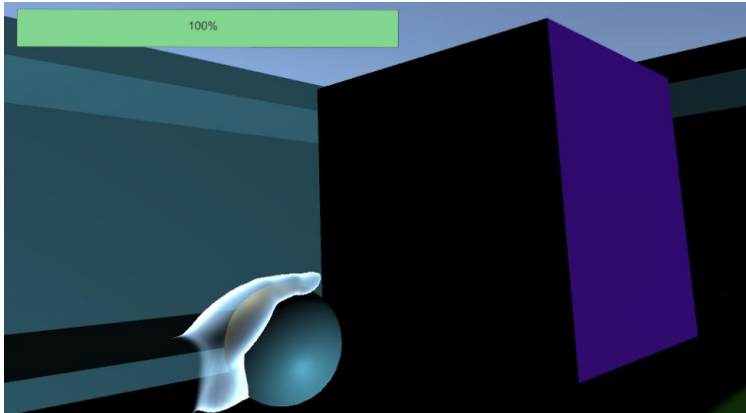
**Gambar 5.10 Tampilan *Client join ke Server***

**Tabel 5.4 Hasil Uji Coba *Multiplayer***

ID	UC-002
Nama	Uji Coba <i>Multiplayer</i> .
Tujuan	Agar bisa memainkan permainan dengan pemain lain.
Kondisi awal	Pemain berada pada <i>screen Network Lobby</i> .
Masukan	Pemain sebagai server memilih tombol Play and Host dan menunggu pemain lain sebagai client untuk bergabung ke lobi kemudian pilih <i>join</i> .
Keluaran yang diharapkan	Terdapat 2 pemain di dalam <i>maze</i> .
Kondisi akhir	2 pemain berhasil bergabung kedalam 1 maze yang sama.
Hasil uji coba	berhasil

### 5.2.3. Pengujian Pergerakan Karakter Pemain

Pengujian dilakukan setelah pemain masuk kedalam *gameplay stage*. Pemain dapat menggerakkan dengan cara berjalan menggunakan *thumbstick* pada oculus untuk menemukan objek *finish*. Berikut hasil pengujian ini dapat dilihat pada Gambar 5.11 dan Tabel 5.5.



**Gambar 5.11 Tampilan pergerakan karakter dan bertemu *finish***

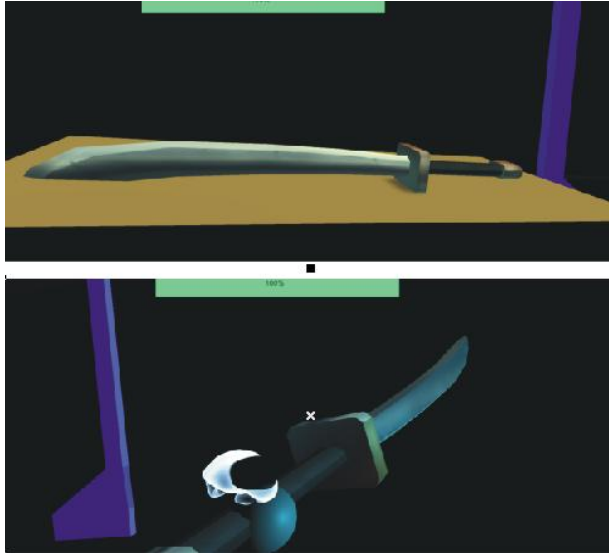
**Tabel 5.5 Hasil Uji Coba Pergerakan Karakter Pemain**

ID	UC-003
Nama	Uji Coba Pergerakan Karakter Pemain.
Tujuan	Pemain dapat menjalankan karakter pemain.
Kondisi awal	Pemain berada didalam <i>maze</i> .
Masukan	Pemain menggerakkan analog pada kontroler.
Keluaran yang diharapkan	Pemain dapat berpindah tempat dari tempat sebelumnya.
Kondisi akhir	Pemain berpindah tempat.
Hasil uji coba	berhasil

#### **5.2.4. Pengujian Interaksi dengan Objek**

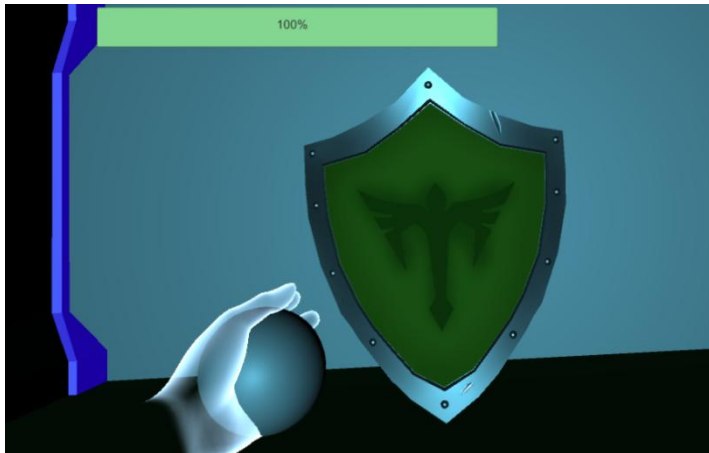
Pada pengujian Interaksi terdapat 2 Skenario, yang pertama memegang senjata, yang kedua adalah mendapatkan *powerup*. Pengujian memegang senjata dimulai ketika pemain berada di dalam *maze* salah satu contohnya adalah berinteraksi dengan *senjata*. Pemain dapat memegang senjata sebagai alat untuk melawan musuh yang akan muncul ditengah permainan.

Berikut hasil pengujian dapat dilihat pada Gambar 5.12 dan Tabel 5.6.



**Gambar 5.12 Tampilan Berinteraksi dengan Objek**

Pengujian *powerup* ini ketika pemain berada dalam *gamescene*. Untuk kondisi awalnya *powerup* akan ada di posisi tertentu. Karakter pemain melewati *powerup* kemudian *powerup* yang dilewati akan menghilang dan pemain mendapatkan kekuatan tambahan dari *powerup*. Seperti pada Gambar 5.13 adalah salah satu contoh *powerup*. Hasil uji coba dapat dilihat pada Tabel 5.6.



**Gambar 5.13 Tampilan menemukan *PowerUp***

**Tabel 5.6 Hasil Uji Coba Interaksi dengan Objek**

ID	UC-004
Nama	Uji Coba Interaksi dengan Objek.
Tujuan	Pemain mengetahui cara berinteraksi dengan objek yang ada di dalam <i>maze</i> .
Skenario 1. (Memegang senjata)	
Kondisi awal	Pemain berada didalam <i>game</i> .
Masukan	Pemain menekan grip pada kontroler ketika akan mengambil objek seperti senjata.
Keluaran yang diharapkan	Pemain dapat memegang senjata.
Kondisi akhir	Pemain memegang senjata.
Skenario 2. (Mendapatkan <i>powerup</i> )	
Kondisi awal	Pemain berada didalam <i>game</i> .
Masukan	Pemain menabrak <i>powerup</i> .
Keluaran yang diharapkan	Pemain diharapkan mendapatkan efek dari <i>powerup</i> .

Kondisi akhir	Pemain mendapatkan efek.
Hasil uji coba	berhasil

### 5.2.5. Pengujian Menang dan Kalah

Pengujian ini dimulai dengan menyelesaikan stage permainan. Saat pemain telah menemukan *finish* yang ada dalam stage dan memenangkan permainan maka *window* atau tampilan menang akan muncul, menandakan pemain telah sukses menyelesaikan stage. Tampilan pemain menang dan tampilan stage menang muncul dapat dilihat pada Gambar 5.14.



**Gambar 5.14 Tampilan Menang**

Pemain akan dinyatakan kalah ketika *health* pemain mencapai 0. Tampilan pengujian pemain kalah karena diserang musuh dan *health* mencapai 0 dapat dilihat pada Gambar 5.15. Hasil uji coba dapat dilihat pada Tabel 5.7.



**Gambar 5.15 Tampilan Kalah**

**Tabel 5.7 Hasil Uji Coba Menang dan Kalah**

ID	UC-005
Nama	Uji Coba Menang dan Kalah.
Skenario 1. (Pemain menemukan <i>finish</i> )	
Kondisi awal	Pemain berada didalam <i>game</i> .
Masukan	Pemain menemukan <i>finish</i> dan menabraknya.
Keluaran yang diharapkan	Notifikasi menang akan muncul.
Kondisi akhir	Notifikasi menang muncul.
Hasil uji coba	berhasil
Skenario 2. ( <i>Health</i> pemain berkurang menjadi 0)	
Kondisi awal	Pemain berada didalam <i>game</i> .
Masukan	Pemain berhadapan dengan musuh dan <i>health</i> pemain mencapai 0.
Keluaran yang diharapkan	Notifikasi kalah akan muncul.
Kondisi akhir	Notifikasi kalah muncul.
Hasil uji coba	berhasil

### 5.2.6. Pengujian Main Menu

Pada pengujian *Main Menu* dilakukan uji coba dengan mengarahkan laser pada tombol-tombol tertentu kemudian menekan *trigger* pada controller Oculus.

Pengujian dimulai ketika pemain telah masuk ke layar *Main Menu* seperti Gambar 5.16. Pemain memilih tombol ‘Play’ untuk masuk kedalam layar *Level Selection* seperti Gambar 5.17. Hasil pengujian dapat dilihat pada Tabel 5.8.



**Gambar 5.16 Antarmuka *Main Menu***





**Gambar 5.17 Tampilan *Level Selection***

**Tabel 5.8 Uji Coba Main Menu**

ID	UC-006
Nama	Uji Coba Main Menu.
Skenario 1. (Pemain memilih tombol Play)	
Kondisi awal	Pemain berada pada <i>main menu screen</i> .
Masukan	Pemain mengarahkan laser pada tombol <i>play</i> kemudian menekan <i>trigger</i> pada kontroler.
Keluaran yang diharapkan	<i>Screen</i> dapat berpindah dari <i>main menu</i> ke <i>level selection</i> .
Kondisi akhir	<i>Screen</i> berpindah ke <i>level selection</i> .
Hasil uji coba	berhasil
Skenario 2. (Pemain memilih tombol How to Play)	
Kondisi awal	Pemain berada pada <i>main menu screen</i> .

Masukan	Pemain mengarahkan laser pada tombol <i>how to play</i> kemudian menekan <i>trigger</i> pada kontroler.
Keluaran yang diharapkan	<i>Screen</i> dapat berpindah dari <i>main menu</i> ke <i>how to play screen</i> .
Kondisi akhir	<i>Screen</i> berpindah ke <i>how to play</i> .
Hasil uji coba	berhasil
Skenario 3. (Pemain memilih tombol About)	
Kondisi awal	Pemain berada pada <i>main menu screen</i> .
Masukan	Pemain mengarahkan laser pada tombol <i>about</i> kemudian menekan <i>trigger</i> pada kontroler.
Keluaran yang diharapkan	<i>Screen</i> dapat berpindah dari <i>main menu</i> ke <i>about screen</i> .
Kondisi akhir	<i>Screen</i> berpindah ke <i>about</i> .
Hasil uji coba	berhasil
Skenario 4. (Pemain memilih tombol Quit)	
Kondisi awal	Pemain berada pada <i>main menu screen</i> .
Masukan	Pemain mengarahkan laser pada tombol <i>quit</i> kemudian menekan <i>trigger</i> pada kontroler.
Keluaran yang diharapkan	Keluar dari <i>game</i> .
Kondisi akhir	<i>Game</i> tertutup.
Hasil uji coba	berhasil

### 5.2.7. Pengujian Level Selection

Pada pengujian *level selection* dilakukan uji coba untuk memilih *level* yang diinginkan oleh pengguna, yaitu dengan cara mengarahkan laser ke salah satu *level* yang ingin dimainkan kemudian menekan *trigger* pada kontroler.

Pengujian dimulai ketika pemain berada dalam *level selection screen* seperti pada Gambar 5.18. Hasil uji coba *level selection* dapat dilihat pada Tabel 5.9.



**Gambar 5.18 Pengujian *Level Selection***

**Tabel 5.9 Hasil Uji Coba *Level Selection***

ID	UC-007
Nama	Uji Coba <i>Level Selection</i> .
Kondisi awal	Pemain berada pada <i>menu level selection</i> .
Masukan	Pemain mengarahkan laser pada salah satu tombol <i>level</i> kemudian menekan <i>trigger</i> pada kontroler.
Keluaran yang diharapkan	<i>Screen</i> dapat berpindah dari <i>level selection</i> ke <i>screen Network Lobby</i> .
Kondisi akhir	<i>Screen</i> berpindah ke <i>Network Lobby</i> .
Hasil uji coba	berhasil

### 5.3. Pengujian Pengguna

Selain melakukan uji coba fungsionalitas, dilakukan juga pengujian yang melibatkan pengguna secara langsung. Pengujian

ini bernilai subjektif yang bertujuan untuk mengetahui tingkat keberhasilan *gameplay* dari sudut pandang pengguna. Penilaian ini bisa didapat dengan meminta penilaian dan tanggapan dari pengguna terhadap aspek-aspek pada *gameplay*.

### 5.3.1. Skenario Pengujian Pengguna

Dalam melakukan pengujian perangkat lunak, pengguna diminta mencoba menggunakan perangkat lunak untuk mencoba semua fungsionalitas dan fitur yang ada. Pengujian aplikasi oleh pengguna dilakukan dengan memberikan informasi terlebih dahulu mengenai aplikasi, kegunaan, dan fitur-fitur yang dimiliki. Setelah informasi tersampaikan, pengguna kemudian diarahkan untuk langsung mencoba aplikasi dengan spesifikasi lingkungan yang sama dengan yang telah diuraikan pada Tabel 5.1.

Jumlah pengguna yang terlibat dalam pengujian perangkat lunak sebanyak 7 orang. Dalam melakukan pengujian, pengguna melakukan percobaan lebih dari satu kali penggunaan untuk masing-masing pengguna. Dalam memberikan penilaian dan tanggapan, pengguna diberikan kuesioner pengujian perangkat lunak. Kuesioner pengujian perangkat lunak ini memiliki beberapa aspek penilaian seputar desain antarmuka, *immersive*, *artificial intelligence*, *minigame*, *gameplay* dan tingkat kenyamanan aplikasi. Nilai yang diberikan memiliki rentang nilai 1 hingga 6 dengan rincian pada Tabel 5.10. Pada bagian akhir terdapat saran untuk perbaikan fitur. Detail kuesioner pengguna dapat dilihat pada Tabel 5.11.

**Tabel 5.10 Rentang Nilai**

No.	Keterangan	Nilai
1	Sangat Tidak Setuju (STS)	1
2	Tidak Setuju (TS)	2
3	Kurang Setuju (KS)	3
4	Cukup Setuju (CS)	4
5	Setuju (S)	5
6	Sangat Setuju (SS)	6

**Tabel 5.11 Kuesioner Pengguna**

No.	Karakteristik Pemain						
1	Apakah anda mengetahui <i>Virtual Reality</i> ?						
2	Apakah anda mengetahui permainan labirin ( <i>maze game</i> )?						
3	Apakah anda pernah bermain permainan labirin?						
4	Apakah anda pernah bermain permainan labirin berbasis <i>Virtual Reality</i> ?						
5	Jika pernah, Bagaimana mendapat anda?						
5	Apakah anda mengetahui <i>Artificial Intelligence</i> ?						
No.	Parameter Antar Muka	STS	TS	KS	CS	S	SS
1.	Aplikasi memiliki tampilan, warna, dan desain yang menarik.						
2.	Aplikasi ini memiliki tata letak tombol, instruksi dan informasi yang mudah dipahami						
No.	Parameter Immersive	STS	TS	KS	CS	S	SS
3.	Anda merasakan sensasi nyata seperti di dalam <i>maze</i> sungguhan						
4.	Anda merasa tertantang untuk segera mencari jalan keluar <i>maze</i>						
5.	Anda merasakan suasana misterius yang ditawarkan oleh permainan						
No.	Parameter <i>Artificial Intelligence</i>	STS	TS	KS	CS	S	SS
6	Musuh dalam permainan ini dapat menemukan anda						
7	Semakin tinggi level,						

	maka semakin cepat musuh menemukan anda						
8	Tingkat kesulitan musuh sesuai dengan level permainan						
No.	Parameter <i>Minigame</i>	STS	TS	KS	CS	S	SS
9	<i>Minigame</i> Legot dapat melatih kemampuan logika anda						
10	<i>Minigame</i> Legot pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai						
11	<i>Minigame</i> Legot pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai						
12	<i>Minigame</i> Legot pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai						
13	<i>Minigame</i> Picture Memory dapat melatih memori anda						
14	<i>Minigame</i> Picture Memory pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai						
15	<i>Minigame</i> Picture Memory pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai						
16	<i>Minigame</i> Picture Memory pada level <i>hard</i> memiliki tingkat						

	kesulitan yang sesuai						
17	<i>Minigame</i> Tap Fast dapat melatih kecepatan refleks anda						
18	<i>Minigame</i> Tap Fast pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai						
19	<i>Minigame</i> Tap Fast pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai						
20	<i>Minigame</i> Tap Fast pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai						
No.	Parameter <i>Gameplay</i>	STS	TS	KS	CS	S	SS
21	Rancangan <i>random maze</i> sudah sesuai dengan tingkat kesulitan levelnya						
22	Saya merasa <i>game</i> ini cocok dimainkan secara <i>multiplayer</i>						
23	Saya merasa dengan bantuan <i>power up</i> , <i>game</i> ini menjadi menarik						
No.	Parameter Kenyamanan	STS	TS	KS	CS	S	SS
24	Aplikasi dapat berjalan lancar tanpa <i>lag</i> dan <i>crash</i>						
25	Kontrol pergerakan <i>player</i> tidak membingungkan.						
26	Saya merasa nyaman						

	selama menggunakan aplikasi ini						
--	---------------------------------	--	--	--	--	--	--

### 5.3.2. Daftar Pengujian Pemain

Pada sub bab ini berisi daftar pengguna yang bertindak sebagai penguji coba *game* yang dibangun. Dalam pengujian ini tidak terdapat kriteria atau keahlian khusus yang harus dimiliki pengguna karena *game* ini ditunjukkan kepada berbagai kalangan pengguna baik yang suka bermain permainan ataupun tidak. Daftar nama penguji aplikasi ini dapat dilihat pada Tabel 5.12.

**Tabel 5.12 Daftar Penguji Permainan**

No.	Nama	Pekerjaan
1	Ghaly Aditya	Mahasiswa
2	Riansyah Pramusti	Mahasiswa
3	Yusuf Dimas Hermawan	Mahasiswa
4	Rahmat Rijal	Mahasiswa
5	Sultan Bonar M	Mahasiswa
6	Ananda Ricky	Mahasiswa
7	Hari Setiawan	Mahasiswa

### 5.3.3. Hasil Pengujian Pengguna

Uji coba yang dilakukan terhadap beberapa pengguna memiliki beberapa aspek yang dipisahkan berdasarkan antarmuka, *immersive*, *artificial intelligence*, *minigame*, *gameplay* dan tingkat kenyamanan aplikasi. Sistem penilaian didasarkan pada skala penghitungan satu sampai enam dimana skala satu menunjukkan nilai terendah dan skala enam menunjukkan skala tertinggi. Penilaian akhir kemudian dilakukan dengan menghitung berapa banyak penguji yang memilih suatu skala tertentu dan kemudian dicari nilai rata-ratanya. Hasil uji coba dipaparkan secara lengkap dengan disertai tabel yang dapat dilihat pada Tabel 5.13 dan Tabel 5.14.



**Tabel 5.13 Hasil Pengujian Pengguna**

No.	Parameter <i>Gameplay</i>	STS	TS	KS	CS	S	SS
1	Rancangan <i>random maze</i> sudah sesuai dengan tingkat kesulitan levelnya				1	3	3
2	Saya merasa <i>game</i> ini cocok dimainkan secara <i>multiplayer</i>		1		1	2	3
3	Saya merasa dengan bantuan <i>power up</i> , <i>game</i> ini menjadi menarik				1	4	2

**Tabel 5.14 Hasil Akhir Pengujian Pengguna**

No	Pernyataan	Rata-Rata	Rata-Rata Total	Nilai Akhir (%)
1	Rancangan <i>random maze</i> sudah sesuai dengan tingkat kesulitan levelnya	5,29	5	83,33
2	Saya merasa <i>game</i> ini cocok dimainkan secara <i>multiplayer</i>	4,57		
3	Saya merasa dengan bantuan <i>power up</i> , <i>game</i> ini menjadi menarik	5,14		

#### 5.3.4. Kritik dan Saran Pengguna

Dalam memberikan penilaian dan tanggapan, pengguna diberikan kuesioner pengujian perangkat lunak. Kuesioner pengujian perangkat lunak ini terdapat bagian kritik dan saran

untuk perbaikan fitur kedepannya. Kritik dan saran pengguna dapat dilihat pada Tabel 5.15.

**Tabel 5.15 Kritik dan Saran Pengguna**

<b>No.</b>	<b>Nama</b>	<b>Kritik dan Saran</b>
1	Ghaly Aditya	Memperlihatkan untuk beberapa detik sebelum permainan dimulai sehingga pemain bisa mengetahui / mengingat jalan yang dituju.
2	Riansyah Pramusti	Rotasi pemain dibuat halus agar tidak memusingkan (controller).
3	Yusuf Dimas Hermawan	Dibuat menjadi lebih nyaman untuk dimainkan agar tidak pusing saat bermain.
4	Rahmat Rijal	Gameplay AVIAR sudah bagus tetapi kalau bisa asset untuk lingkungan gamenya diperbagus lagi agar membuat pemain tertarik dan merasa nyaman.
5	Sultan Bonar M	Kecepatan saat memakai boots terlalu cepat, bikin pusing. Saran dipelankan (jangan ekstrem).
6	Ananda Ricky	Gamenya buat pusing, berikan penawaran pusing.
7	Hari Setiawan	Pergerakan pemain terlalu cepat sehingga meninggalkan musuh terlalu jauh. Sound effect kurang menarik.

#### **5.4. Evaluasi Pengujian**

Pada sub bab ini akan dibahas mengenai evaluasi terhadap hasil pengujian yang telah dilakukan.

Dari hasil uji fungsionalitas pada Tabel 5.3 sampai dengan Tabel 5.9 didapatkan bahwa *gameplay* permainan AVIAR dapat berjalan sesuai dengan skenario dan keluaran yang diharapkan. Dari data hasil pengujian pengguna pada Tabel 5.13 dan 5.14 didapatkan bahwa masing-masing *minigame* mendapatkan respon baik dari pengguna walaupun masih ada yang harus diperbaiki dan ditambah yang bisa dilihat dari Tabel 5.15 mengenai kritik dan saran.

*[Halaman ini sengaja dikosongkan]*

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

#### **6.1. Kesimpulan**

Dari Proses penerapan algoritma *Recursive Division* untuk perancangan *maze* pada permainan AVIAR dapat diambil kesimpulan sebagai berikut:

1. Untuk membangkitkan *maze* secara acak menggunakan algoritma *recursive division* bermula dengan pembuatan ruang seluas kolom x baris, kemudian memasang dinding secara random secara horisontal atau vertikal dan dinding diberi lubang pada salah satu titik secara random untuk menghubungkan antar ruangan, kemudian mengulangi langkah tersebut hingga ukuran ruangan menjadi minimum.
2. Pengaturan *multiplayer* pada permainan AVIAR menggunakan *network lobby manager* dapat berjalan dengan baik.
3. Permainan AVIAR berhasil dirancang dengan baik. Rancangan *gameplay* permainan yang ditetapkan dan juga fungsionalitas dalam permainan dapat berjalan dengan sebagaimana mestinya.

#### **6.2. Saran**

Saran yang diberikan terkait pengembangan pada Tugas Akhir ini berdasarkan perancangan, implementasi, dan uji coba adalah:

1. Membuat pergerakan pemain menjadi lebih halus supaya pemain tidak menjadi pusing saat memainkan.
2. Pergerakan pemain dan *powerup speedup* terlalu cepat.

3. Fitur untuk melihat keseluruhan *maze*.
4. Perbaiki *asset* dan *soundeffect* biar lebih menarik.

## DAFTAR PUSTAKA

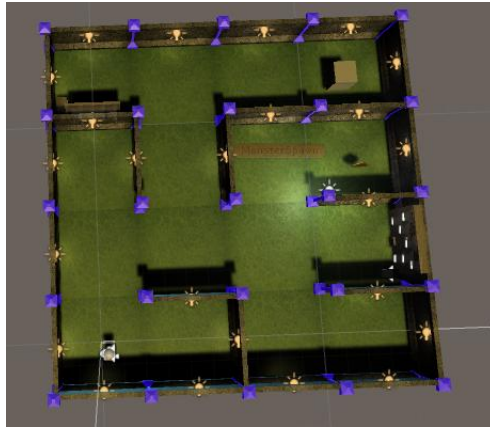
- [1] Rucker, Rudy. (2003). *Software Engineering and Computer Games*. Britain: Dorset Press.
- [2] Dix, A., Finlay, J., Abowd, G. D., & Beale, R. (2004). *Human-Computer Interaction (Third Edition)*. Harlow: Pearson Education Limited.
- [3] Unity Technologies. *CREATE THE GAMES YOU LOVE WITH UNITY*. [Online]. Available: <http://unity3d.com/unity>. [Diakses 25 Mei 2018].
- [4] Hejlsberg, A. (2008). *The A-Z of Programming Languages*.
- [5] R. Pérez Fernández dan V. Alonso, "Virtual Reality in a shipbuilding environment," *Adv. Eng. Softw.*, vol. 81, hal. 30–40, Mar 2015.
- [6] Microsoft, "Microsoft studio- development tools," Microsoft, [Online]. Available: <http://www.visualstudio.com/>. [Diakses 25 Mei 2018]
- [7] Oculus, "Kickstarter," Juni 2015. [Online]. Available: <https://www.kickstarter.com/projects/1523379957/oculus-rift-step-into-the-game>. [Diakses 25 Mei 2018].
- [8] Oculus, "Oculus," 2018. [Online]. Available: <https://www.oculus.com/accessories/>. [Diakses 25 Mei 2018].
- [9] Unity, "Network Lobby Manager," 2018. [Online]. Available: <https://docs.unity3d.com/Manual/class-NetworkLobbyManager.html>. [Diakses 2 Juli 2018]
- [10] Astrolog. *Perfect Maze Creation Algorithms*. (2011, Maret 1). [Online]. Available:

<http://www.astrolog.org/labyrnth/algrithm.htm#perfect>. [Diakses 15 July 2018].

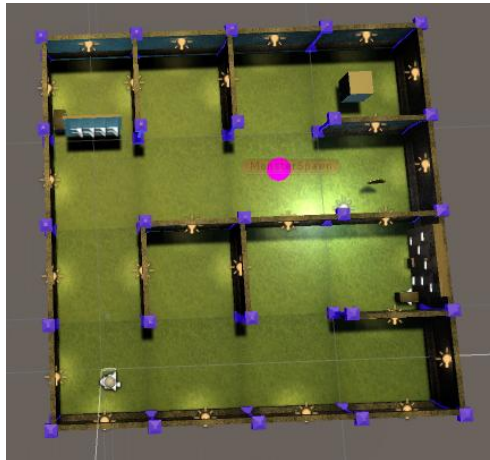


## LAMPIRAN

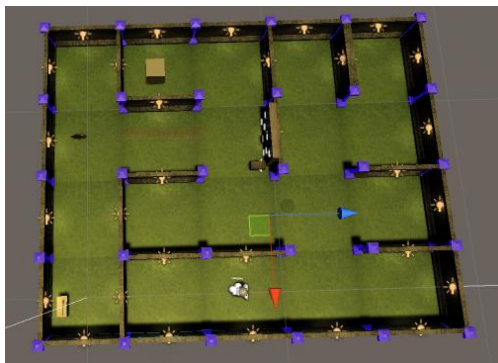
### A. Screenshot *Maze* setiap *Level*



**Gambar A. 1** *Maze Level 1A*



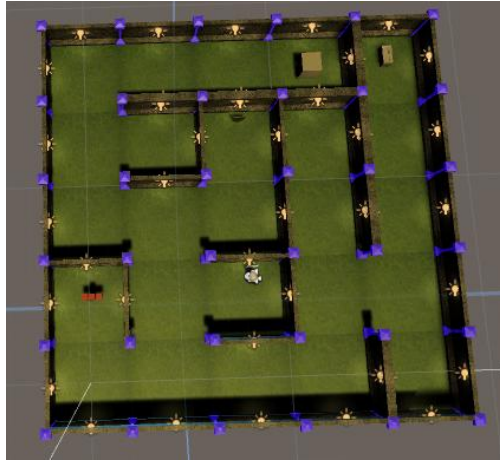
**Gambar A. 2** *Maze Level 1B*



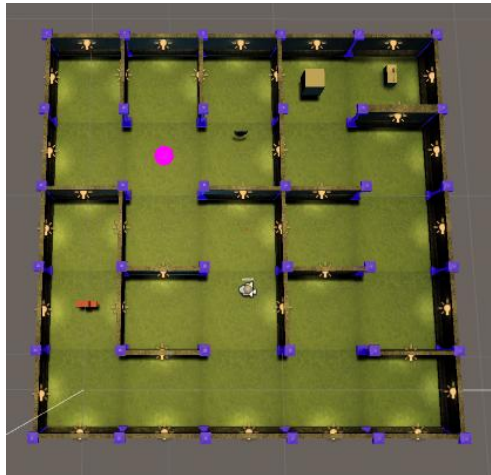
**Gambar A. 3** *Maze Level 2A*



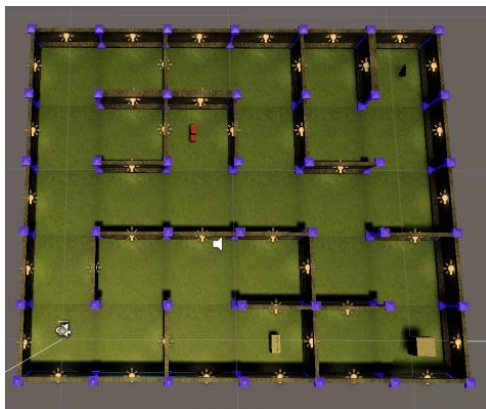
**Gambar A. 4** *Maze Level 2B*



**Gambar A. 5** *Maze Level 3A*



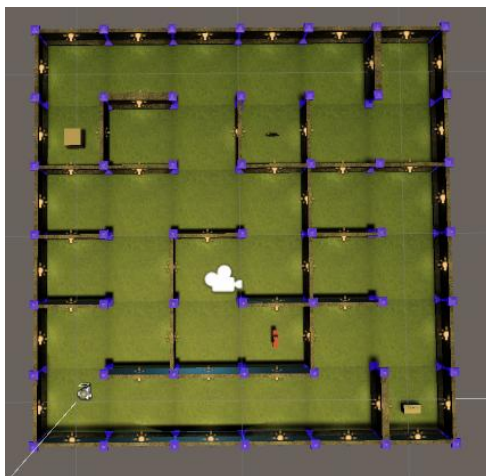
**Gambar A. 6** *Maze Level 3B*



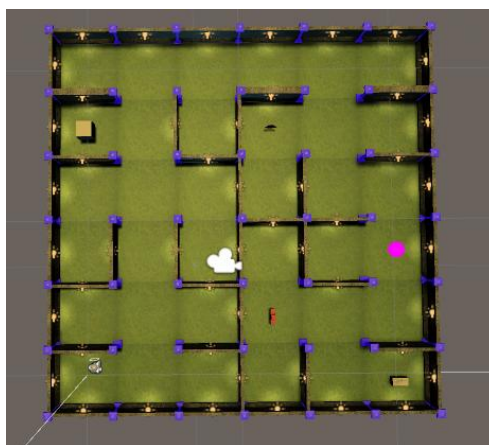
**Gambar A. 7** *Maze Level 4A*



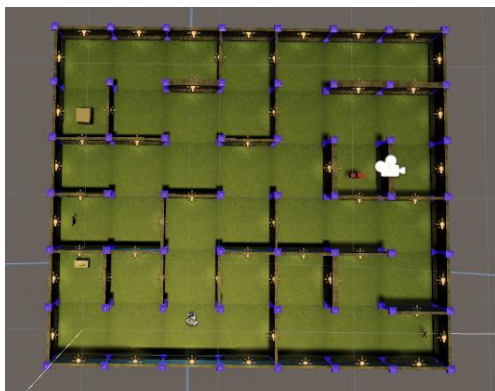
**Gambar A. 8** *Maze Level 4B*



**Gambar A. 9 Maze Level 5A**



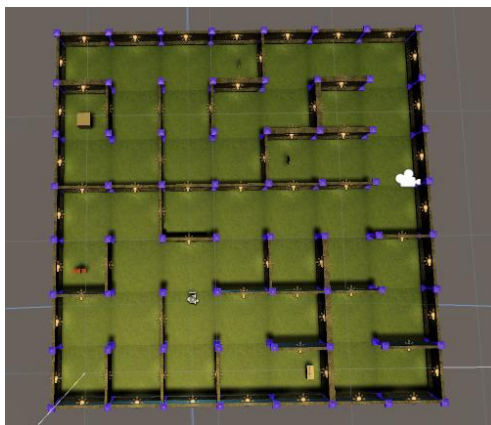
**Gambar A. 10 Maze Level 5B**



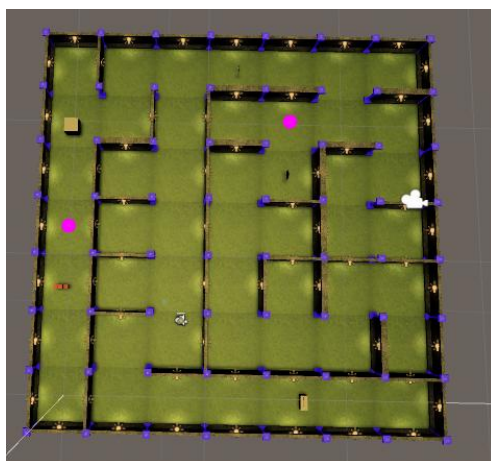
**Gambar A. 11** *Maze Level 6A*



**Gambar A. 12** *Maze Level 6B*



**Gambar A. 13 *Maze Level 7B***



**Gambar A. 14 *Maze Level 7B***



**Gambar A. 15** *Maze Level 8A*



**Gambar A. 16** *Maze Level 8B*





**Gambar A. 17** *Maze Level 9A*



**Gambar A. 18** *Maze Level 9B*



**Gambar A. 19** *Maze Level 10A*



**Gambar A. 20** *Maze Level 10B*



**Gambar A. 21** *Maze Level 11A*



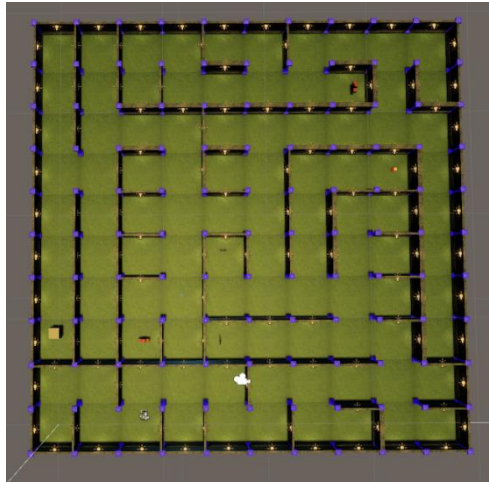
**Gambar A. 22** *Maze Level 11B*



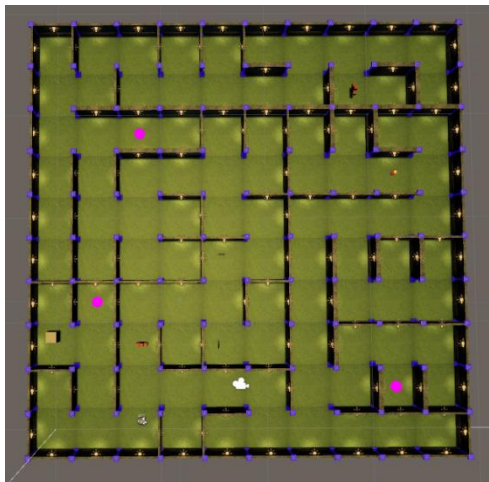
**Gambar A. 23** *Maze Level 12A*



**Gambar A. 24** *Maze Level 12B*



**Gambar A. 25** *Maze Level 13A*



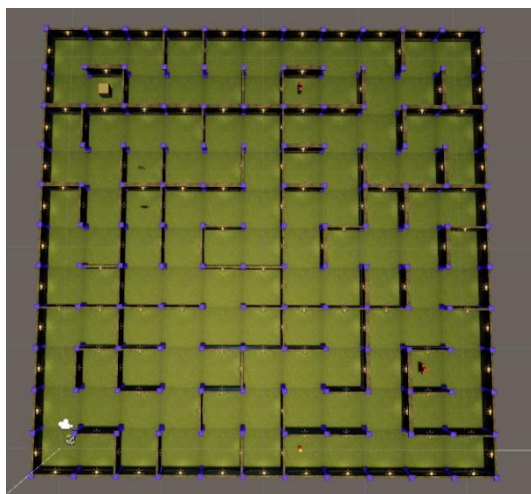
**Gambar A. 26** *Maze Level 13B*



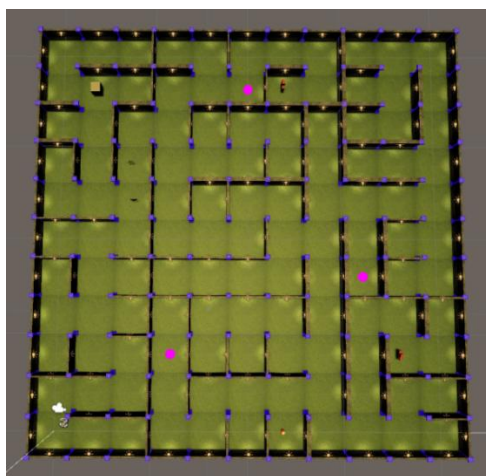
**Gambar A. 27** *Maze Level 14A*



**Gambar A. 28** *Maze Level 14B*



**Gambar A. 29 Maze Level 15A**



**Gambar A. 30 Maze Level 15B**



## B. Hasil Kuesioner

**Kuesioner Tugas Akhir "Game AVIAR"**

5114100065 – Argegi Yudhistira  
5114100066 – Vinsensia S. Zega  
5114100124 – Andar Rizqi

Surabaya, 09 Juli 2018

**IDENTITAS RESPONDEN**

Nama Lengkap : Heri Selwyn  
Pekerjaan : Mahasiswa  
Usia : 22 tahun

**A. KARAKTERISTIK RESPONDEN**

Isilah pertanyaan di bawah ini dengan menggunakan tanda silang (X)

- Apakah anda mengetahui *Virtual Reality* ?  
☒ Iya, Tahu      ☐ Tidak Tahu
- Apakah anda mengetahui permainan labirin (*maze game*) ?  
☒ Iya, Tahu      ☐ Tidak Tahu
- Apakah anda pernah bermain permainan labirin ?  
☒ Iya, Pernah      ☐ Tidak Pernah
- Apakah anda pernah bermain permainan labirin berbasis *Virtual Reality* ?  
 a. Iya, Pernah      ☒ Tidak Pernah  
 Jika Pernah, bagaimana pendapat anda? \_\_\_\_\_
- Apakah anda mengetahui *Artificial Intelligence* ?  
☒ Iya, Pernah      ☐ Tidak Tahu

**B. PENILAIAN TERHADAP APLIKASI**

Isilah pertanyaan di bawah ini dengan menggunakan tanda centang (✓)

Keterangan: STS = Sangat setuju, S = Setuju, CS = Cukup setuju, TS = Tidak Setuju, STS = Sangat tidak setuju

No	Parameter	STS	TS	KS	CS	S	SS
1.	Parameter <i>Antar Muka</i> Aplikasi memiliki tampilan, warna, dan desain yang menarik.					✓	
2.	Aplikasi ini memiliki tata letak tombol, instruksi dan informasi yang mudah dipahami					✓	
3.	Parameter <i>Immersive</i> Anda merasakan sensasi nyata seperti di dalam <i>maze</i> sungguhan						✓
4.	Anda merasa tertantang untuk segera mencari jalan keluar <i>maze</i>						✓
5.	Anda merasakan suasana misterius yang ditawarkan oleh permainan				✓		
6.	Parameter <i>Artificial Intelligence</i> Musuh dalam permainan ini dapat menemukan anda					✓	
7.	Semakin tinggi level, maka semakin cepat musuh menemukan anda					✓	
8.	Tingkat kesulitan musuh sesuai dengan level permainan					✓	
9.	Parameter <i>Mini-game</i> <i>Mini-game</i> Legot dapat melatih kemampuan logika						✓



	anda							
10	<i>Minigame</i> Legot pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai			✓				
11	<i>Minigame</i> Legot pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓		
12	<i>Minigame</i> Legot pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓		
13	<i>Minigame</i> Picture Memory dapat melatih memori anda							✓
14	<i>Minigame</i> Picture Memory pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai							✓
15	<i>Minigame</i> Picture Memory pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai							✓
16	<i>Minigame</i> Picture Memory pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai							✓
17	<i>Minigame</i> Tap Fast dapat melatih kecepatan refleks anda					✓		
18	<i>Minigame</i> Tap Fast pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai					✓		
19	<i>Minigame</i> Tap Fast pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓		
20	<i>Minigame</i> Tap Fast pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai							✓
No.	Parameter <i>Gameplay</i>	STS	TS	KS	CS	S	SS	
21	Rancangan <i>random maze</i> sudah sesuai dengan tingkat kesulitan levelnya							✓
22	Saya merasa <i>game</i> ini cocok dimainkan secara <i>multiplayer</i>					✓		
23	Saya merasa dengan bantuan <i>power up</i> , <i>game</i> ini menjadi menarik							✓
No.	Parameter Kenyamanan	STS	TS	KS	CS	S	SS	
24	Aplikasi dapat berjalan lancar tanpa <i>lag</i> dan <i>crash</i>							✓
25	Kontrol pergerakan <i>player</i> tidak membingungkan							✓
26	Saya merasa nyaman selama bermain permainan ini							✓

### C. KRITIK DAN SARAN

.....

.....

.....



### Kuesioner Tugas Akhir "Game AVIAR"

5114100065 – Anggit Yudhistira  
5114100066 – Vimsensia S. Zega  
5114100124 – Anfar Rizqi

#### IDENTITAS RESPONDEN

Nama Lengkap : Riansya Pamusti  
Pekerjaan : Mahasiswa  
Usia : 22

Surabaya, 9 Juli 2018

*Riansya*

#### A. KARAKTERISTIK RESPONDEN

Isilah pertanyaan di bawah ini dengan menggunakan tanda silang (X)

- Apakah anda mengetahui *Virtual Reality* ?  
☒ Iya, Tahu      b. Tidak Tahu
- Apakah anda mengetahui permainan labirin (*maze game*) ?  
☒ Iya, Tahu      b. Tidak Tahu
- Apakah anda pernah bermain permainan labirin ?  
☒ Iya, Pernah      b. Tidak Pernah
- Apakah anda pernah bermain permainan labirin berbasis *Virtual Reality* ?  
a. Iya, Pernah      ☒ Tidak Pernah  
Jika Pernah, bagaimana pendapat anda?
- Apakah anda mengetahui *Artificial Intelligence* ?  
☒ Iya, Pernah      b. Tidak Tahu

#### B. PENILAIAN TERHADAP APLIKASI

Isilah pertanyaan di bawah ini dengan menggunakan tanda centang (✓)

Keterangan							
SS	= Sangat setuju	S	= Setuju	CS	= Cukup setuju		
KS	= Kurang setuju	TS	= Tidak Setuju	STS	= Sangat tidak setuju		
No.	Parameter Antar Muka	STS	TS	KS	CS	S	SS
1.	Aplikasi memiliki tampilan, warna, dan desain yang menarik					✓	
2.	Aplikasi ini memiliki tata letak tombol, instruksi dan informasi yang mudah dipahami					✓	
No.	Parameter Immersive	STS	TS	KS	CS	S	SS
3.	Anda merasakan sensasi nyata seperti di dalam <i>maze</i> sungguhan						✓
4.	Anda merasa tertantang untuk segera mencari jalan keluar <i>maze</i>						✓
5.	Anda merasakan suasana misterius yang ditawarkan oleh permainan					✓	
No.	Parameter Artificial Intelligence	STS	TS	KS	CS	S	SS
6.	Musuh dalam permainan ini dapat menemukan anda					✓	
7.	Semakin tinggi level, maka semakin cepat musuh menemukan anda					✓	
8.	Tingkat kesulitan musuh sesuai dengan level permainan				✓		
No.	Parameter Minigame	STS	TS	KS	CS	S	SS
9.	<i>Minigame</i> Legot dapat melatih kemampuan logika						✓

	anda								
10	Minigame Legot pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai						✓		
11	Minigame Legot pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai							✓	
12	Minigame Legot pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai							✓	
13	Minigame Picture Memory dapat melatih memori anda							✓	
14	Minigame Picture Memory pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai							✓	
15	Minigame Picture Memory pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai						✓		
16	Minigame Picture Memory pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai						✓		
17	Minigame Tap Fast dapat melatih kecepatan refleks anda						✓		
18	Minigame Tap Fast pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai						✓		
19	Minigame Tap Fast pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai						✓		
20	Minigame Tap Fast pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai						✓		
No	Parameter Gameplay	STS	TS	KS	CS	S	SS		
21	Rancangan <i>random maze</i> sudah sesuai dengan tingkat kesulitan levelnya					✓			
22	Saya merasa <i>game</i> ini cocok dimainkan secara <i>multiplayer</i>							✓	
23	Saya merasa dengan bantuan <i>power up</i> , <i>game</i> ini menjadi menarik					✓			
No	Parameter Kenyamanan	STS	TS	KS	CS	S	SS		
24	Aplikasi dapat berjalan lancar tanpa <i>lag</i> dan <i>crash</i>					✓		✓	
25	Kontrol pergerakan <i>player</i> tidak membingungkan					✓			
26	Saya merasa nyaman selama bermain permainan ini					✓			

## C. KRITIK DAN SARAN

Rotasi pemain dibuat halus agar tidak membingungkan (Controller)



### Kuesioner Tugas Akhir "Game AVIAR"

5114100065 - Anggit Yudhistira  
5114100066 - Vinsensia S. Zega  
5114100124 - Aufar Rizqi

#### IDENTITAS RESPONDEN

Nama Lengkap : Kauf Duce H  
Pekerjaan : Mahasiswa  
Usia : 21

Surabaya, 9 Jul 2018

#### A. KARAKTERISTIK RESPONDEN

Isilah pertanyaan di bawah ini dengan menggunakan tanda silang (X)

- Apakah anda mengetahui *Virtual Reality* ?  
☒ a. Iya, Tahu      b. Tidak Tahu
- Apakah anda mengetahui permainan labirin (*maze game*) ?  
☒ a. Iya, Tahu      b. Tidak Tahu
- Apakah anda pernah bermain permainan labirin ?  
☒ a. Iya, Pernah      b. Tidak Pernah
- Apakah anda pernah bermain permainan labirin berbasis *Virtual Reality* ?  
☐ a. Iya, Pernah      ☒ b. Tidak Pernah  
 Jika Pernah, bagaimana pendapat anda?
- Apakah anda mengetahui *Artificial Intelligence* ?  
☒ a. Iya, Pernah      b. Tidak Tahu

#### B. PENILAIAN TERHADAP APLIKASI

Isilah pertanyaan di bawah ini dengan menggunakan tanda centang (✓)

Keterangan:							
SS = Sangat setuju		S = Setuju		CS = Cukup setuju		STS = Sangat tidak setuju	
KS = Kurang setuju		TS = Tidak Setuju					
No	Parameter Antar Muka	STS	TS	KS	CS	S	SS
1.	Aplikasi memiliki tampilan, warna, dan desain yang menarik.					<input checked="" type="checkbox"/>	
2.	Aplikasi ini memiliki tata letak tombol, instruksi dan informasi yang mudah dipahami				<input checked="" type="checkbox"/>		
No	Parameter Immersive	STS	TS	KS	CS	S	SS
3.	Anda merasakan sensasi nyata seperti di dalam <i>maze</i> sungguhan						<input checked="" type="checkbox"/>
4.	Anda merasa tertantang untuk segera mencari jalan keluar <i>maze</i>					<input checked="" type="checkbox"/>	
5.	Anda merasakan suasana misterius yang ditawarkan oleh permainan				<input checked="" type="checkbox"/>		
No	Parameter Artificial Intelligence	STS	TS	KS	CS	S	SS
6.	Musuh dalam permainan ini dapat menemukan anda					<input checked="" type="checkbox"/>	
7.	Semakin tinggi level, maka semakin cepat musuh menemukan anda				<input checked="" type="checkbox"/>		
8.	Tingkat kesulitan musuh sesuai dengan level permainan					<input checked="" type="checkbox"/>	
No	Parameter Minigame	STS	TS	KS	CS	S	SS
9.	<i>Minigame</i> Legot dapat melatih kemampuan logika					<input checked="" type="checkbox"/>	

	anda						
10	Minigame Legot pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai					✓	
11	Minigame Legot pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓	
12	Minigame Legot pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓	
13	Minigame Picture Memory dapat melatih memori anda					✓	
14	Minigame Picture Memory pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai					✓	
15	Minigame Picture Memory pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓	
16	Minigame Picture Memory pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓	
17	Minigame Tap Fast dapat melatih kecepatan refleks anda						✓
18	Minigame Tap Fast pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai					✓	
19	Minigame Tap Fast pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓	
20	Minigame Tap Fast pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓	
No.	Parameter <i>Gameplay</i>	STS	TS	KS	CS	S	SS
21	Rancangan <i>random maze</i> sudah sesuai dengan tingkat kesulitan levelnya						✓
22	Saya merasa <i>game</i> ini cocok dimainkan secara <i>multiplayer</i>				✓		
23	Saya merasa dengan bantuan <i>power up</i> , <i>game</i> ini menjadi menarik					✓	
No.	Parameter <i>Kenyamanan</i>	STS	TS	KS	CS	S	SS
24	Aplikasi dapat berjalan lancar tanpa <i>lag</i> dan <i>crash</i>						✓
25	Kontrol pergerakan <i>player</i> tidak membingungkan					✓	
26	Saya merasa nyaman selama bermain permainan ini				✓		

### C. KRITIK DAN SARAN

Dikawatir menjadi lebih ngeon untuk dimainkan agar tidak pusing saat bermain



### Kuesioner Tugas Akhir "Game AVIAR"

5114100065 – Anggit Yudhistira  
5114100066 – Vinsentia S. Zega  
5114100124 – Anfar Rizqi

#### IDENTITAS RESPONDEN

Nama Lengkap : Evan Rivan M.  
Pekerjaan : Manajemen  
Usia : 21

Surabaya, 7 Juli 2018

#### A. KARAKTERISTIK RESPONDEN

Isilah pertanyaan di bawah ini dengan menggunakan tanda silang (X)

- Apakah anda mengetahui *Virtual Reality* ?  
☒ a. Iya, Tahu      b. Tidak Tahu
- Apakah anda mengetahui permainan labirin (*maze game*) ?  
☒ a. Iya, Tahu      b. Tidak Tahu
- Apakah anda pernah bermain permainan labirin ?  
☒ a. Iya, Pernah      b. Tidak Pernah
- Apakah anda pernah bermain permainan labirin berbasis *Virtual Reality* ?  
☒ a. Iya, Pernah      ☒ b. Tidak Pernah  
 Jika Pernah, bagaimana pendapat anda?
- Apakah anda mengetahui *Artificial Intelligence* ?  
☒ a. Iya, Pernah      b. Tidak Tahu

#### B. PENILAIAN TERHADAP APLIKASI

Isilah pertanyaan di bawah ini dengan menggunakan tanda centang (✓)

Keterangan :		S = Setuju		CS = Cukup setuju		SS = Sangat setuju	
SS = Sangat setuju		S = Setuju		CS = Cukup setuju		SS = Sangat setuju	
KS = Kurang setuju		TS = Tidak Setuju		STS = Sangat tidak setuju			
No	Parameter Antar Muka	STS	TS	KS	CS	S	SS
1.	Aplikasi memiliki tampilan, warna, dan desain yang menarik.					✓	
2.	Aplikasi ini memiliki tata letak tombol, instruksi dan informasi yang mudah dipahami					✓	
No	Parameter Immersive	STS	TS	KS	CS	S	SS
3.	Anda merasakan sensasi nyata seperti di dalam <i>maze</i> sungguhan					✓	
4.	Anda merasa tertantang untuk segera mencari jalan keluar <i>maze</i>					✓	
5.	Anda merasakan suasana misterius yang ditawarkan oleh permainan					✓	
No	Parameter Artificial Intelligence	STS	TS	KS	CS	S	SS
6.	Musuh dalam permainan ini dapat menemukan anda				✓		
7.	Semakin tinggi level, maka semakin cepat musuh menemukan anda				✓		
8.	Tingkat kesulitan musuh sesuai dengan level permainan				✓		
No	Parameter Minigame	STS	TS	KS	CS	S	SS
9.	Minigame Legot dapat melatih kemampuan logika					✓	



	anda								
10	Minigame Legot pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai								✓
11	Minigame Legot pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai							✓	
12	Minigame Legot pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai							✓	
13	Minigame Picture Memory dapat melatih memori anda							✓	
14	Minigame Picture Memory pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai							✓	
15	Minigame Picture Memory pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai							✓	
16	Minigame Picture Memory pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai							✓	
17	Minigame Tap Fast dapat melatih kecepatan refleks anda					✓			
18	Minigame Tap Fast pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai							✓	
19	Minigame Tap Fast pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai							✓	
20	Minigame Tap Fast pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai							✓	
No	Parameter <i>Gameplay</i>	STS	TS	KS	CS	S	SS		
21	Rancangan <i>random maze</i> sudah sesuai dengan tingkat kesulitan levelnya							✓	
22	Saya merasa <i>game</i> ini cocok dimainkan secara <i>multiplayer</i>							✓	
23	Saya merasa dengan bantuan <i>power up</i> , <i>game</i> ini menjadi menarik							✓	
No	Parameter Kenyamanan	STS	TS	KS	CS	S	SS		
24	Aplikasi dapat berjalan lancar tanpa <i>lag</i> dan <i>crash</i>								✓
25	Kontrol pergerakan <i>player</i> tidak membingungkan							✓	
26	Saya merasa nyaman selama bermain permainan ini					✓			

## C. KRITIK DAN SARAN

kecepatan saat memakai boots terlalu cepat bikin  
 pusing.  
 saran diperlambat (jangan ekstrem)



### Kuesioner Tugas Akhir "Game AVIAR"

5114100065 – Anggit Yudhistira  
5114100066 – Vimsensia S. Zega  
5114100124 – Auzar Rizqi

#### IDENTITAS RESPONDEN

Nama Lengkap : Ghaly Aditya  
Pekerjaan : Mahasiswa  
Usia : 19

Surabaya, 9 Juli 2018

#### A. KARAKTERISTIK RESPONDEN

Isilah pertanyaan di bawah ini dengan menggunakan tanda silang (X)

- Apakah anda mengetahui *Virtual Reality* ?  
☒ a. Iya, Tahu      b. Tidak Tahu
- Apakah anda mengetahui permainan labirin (*maze game*) ?  
☒ a. Iya, Tahu      b. Tidak Tahu
- Apakah anda pernah bermain permainan labirin ?  
 a. Iya, Pernah      ☒ b. Tidak Pernah
- Apakah anda pernah bermain permainan labirin berbasis *Virtual Reality* ?  
 a. Iya, Pernah      ☒ b. Tidak Pernah  
 Jika Pernah, bagaimana pendapat anda? .....
- Apakah anda mengetahui *Artificial Intelligence* ?  
☒ a. Iya, Pernah      b. Tidak Tahu

#### B. PENILAIAN TERHADAP APLIKASI

Isilah pertanyaan di bawah ini dengan menggunakan tanda centang (✓)

Keterangan :		S = Setuju		CS = Cukup setuju		STS = Sangat tidak setuju	
SS = Sangat setuju		TS = Tidak Setuju					
KS = Kurang setuju							
No	Parameter Antar Muka	STS	TS	KS	CS	S	SS
1.	Aplikasi memiliki tampilan, warna, dan desain yang menarik.			✓			
2.	Aplikasi ini memiliki tata letak tombol, instruksi dan informasi yang mudah dipahami						✓
No	Parameter Immersive	STS	TS	KS	CS	S	SS
3.	Anda merasakan sensasi nyata seperti di dalam <i>maze</i> sungguhan				✓		
4.	Anda merasa tertantang untuk segera mencari jalan keluar <i>maze</i>					✓	
5.	Anda merasakan suasana misterius yang ditawarkan oleh permainan			✓			
No	Parameter Artificial Intelligence	STS	TS	KS	CS	S	SS
6.	Musuh dalam permainan ini dapat menemukan anda					✓	
7.	Semakin tinggi level, maka semakin cepat musuh menemukan anda						✓
8.	Tingkat kesulitan musuh sesuai dengan level permainan				✓		
No	Parameter Minigame	STS	TS	KS	CS	S	SS
9.	Minigame Legot dapat melatih kemampuan logika					✓	



	anda						
10	Minigame Legot pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai					✓	
11	Minigame Legot pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai				✓		
12	Minigame Legot pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai				✓		
13	Minigame Picture Memory dapat melatih memori anda			✓			
14	Minigame Picture Memory pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai				✓		
15	Minigame Picture Memory pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓	
16	Minigame Picture Memory pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai				✓		
17	Minigame Tap Fast dapat melatih kecepatan refleks anda			✓			
18	Minigame Tap Fast pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai			✓			
19	Minigame Tap Fast pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai				✓		
20	Minigame Tap Fast pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓	
No.	Parameter Gameplay	STS	TS	KS	CS	S	SS
21	Rancangan <i>random maze</i> sudah sesuai dengan tingkat kesulitan levelnya				✓		
22	Saya merasa <i>game</i> ini cocok dimainkan secara <i>multiplayer</i>		✓				
23	Saya merasa dengan bantuan <i>power up</i> , <i>game</i> ini menjadi menarik					✓	
No.	Parameter Kenyamanan	STS	TS	KS	CS	S	SS
24	Aplikasi dapat berjalan lancar tanpa <i>lag</i> dan <i>crash</i>				✓		
25	Kontrol pergerakan <i>player</i> tidak membingungkan						✓
26	Saya merasa nyaman selama bermain permainan ini			✓			

## C. KRITIK DAN SARAN

Memperlihatkan utk bbrp detik sebelum permainan dimulai  
 Sehingga pemain bisa mengetahui / mengingat jalan yg diruju



ITS  
Institut Teknologi Sepuluh Nopember

### Kuesioner Tugas Akhir "Game AVIAR"

5114100065 - Anggit Yudhistira

5114100066 - Vinsensia S. Zega

5114100124 - Aulfar Rizqi

#### IDENTITAS RESPONDEN

Nama Lengkap : Ananda Ricky  
Pekerjaan : Mahasiswa Sem 8  
Usia : 22 tahun

Surabaya, 9 Juli 2018

#### A. KARAKTERISTIK RESPONDEN

Isilah pertanyaan di bawah ini dengan menggunakan tanda silang (X)

1. Apakah anda mengetahui Virtual Reality ?  
☒ Iya, Tahu      b. Tidak Tahu
2. Apakah anda mengetahui permainan labirin (maze game) ?  
☒ Iya, Tahu      b. Tidak Tahu
3. Apakah anda pernah bermain permainan labirin ?  
☒ Iya, Pernah      b. Tidak Pernah
4. Apakah anda pernah bermain permainan labirin berbasis Virtual Reality ?  
☒ Iya, Pernah      b. Tidak Pernah  
Jika Pernah, bagaimana pendapat anda?  
Bagus, tapi ngorok sedikit
5. Apakah anda mengetahui Artificial Intelligence ?  
☒ Iya, Pernah      b. Tidak Tahu

#### B. PENILAIAN TERHADAP APLIKASI

Isilah pertanyaan di bawah ini dengan menggunakan tanda centang (✓)

Keterangan :		S	TS	KS	CS	S	SS
SS = Sangat setuju		S = Setuju	TS = Tidak Setuju	KS = Cukup setuju	CS = Sangat tidak setuju		
KS = Kurang setuju							
No	Parameter Antar Muka	STS	TS	KS	CS	S	SS
1.	Aplikasi memiliki tampilan, warna, dan desain yang menarik.					✓	
2.	Aplikasi ini memiliki tata letak tombol, instruksi dan informasi yang mudah dipahami				✓	✓	
No	Parameter Immersive	STS	TS	KS	CS	S	SS
3.	Anda merasakan sensasi nyata seperti di dalam maze sungguhan					✓	
4.	Anda merasa tertantang untuk segera mencari jalan keluar maze				✓		
5.	Anda merasakan suasana misterius yang ditawarkan oleh permainan				✓		
No	Parameter Artificial Intelligence	STS	TS	KS	CS	S	SS
6.	Musuh dalam permainan ini dapat menemukan anda			✓			
7.	Semakin tinggi level, maka semakin cepat musuh menemukan anda			✓			
8.	Tingkat kesulitan musuh sesuai dengan level permainan		✓				
No	Parameter Minigame	STS	TS	KS	CS	S	SS
9.	Minigame Legot dapat melatih kemampuan logika					✓	

	anda						
10	Minigame Legot pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai					✓	
11	Minigame Legot pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓	
12	Minigame Legot pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓	
13	Minigame Picture Memory dapat melatih memori anda					✓	
14	Minigame Picture Memory pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai					✓	
15	Minigame Picture Memory pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓	
16	Minigame Picture Memory pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓	
17	Minigame Tap Fast dapat melatih kecepatan refleks anda					✓	
18	Minigame Tap Fast pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai					✓	
19	Minigame Tap Fast pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓	
20	Minigame Tap Fast pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓	
No.	Parameter Gameplay	STS	TS	KS	CS	S	SS
21	Rancangan <i>random maze</i> sudah sesuai dengan tingkat kesulitan levelnya					✓	
22	Saya merasa <i>game</i> ini cocok dimainkan secara <i>multiplayer</i>						✓
23	Saya merasa dengan bantuan <i>power up</i> , <i>game</i> ini menjadi menarik				✓		
No.	Parameter Kenyamanan	STS	TS	KS	CS	S	SS
24	Aplikasi dapat berjalan lancar tanpa <i>lag</i> dan <i>crash</i>					✓	
25	Kontrol pergerakan <i>player</i> tidak membingungkan			✓			
26	Saya merasa nyaman selama bermain permainan ini		✓				

## C. KRITIK DAN SARAN

Comment: *game ini bagus. Berikan fitur power up.*



ITS  
Institut Teknologi Sepuluh Nopember

## Kuesioner Tugas Akhir "Game AVIAR"

5114100065 – Anggit Yudhistira  
5114100066 – Vinsensia S. Zega  
5114100124 – Aulaz Rizqi

### IDENTITAS RESPONDEN

Nama Lengkap : Rahmat Rival  
Pekerjaan : Mahasiswa  
Usia : 22 tahun

Surabaya, 28 Juli 2018

Rahmat Rival  
Rival

### A. KARAKTERISTIK RESPONDEN

Isilah pertanyaan di bawah ini dengan menggunakan tanda silang (X)

- Apakah anda mengetahui *Virtual Reality* ?  
☒ Iya, Tahu      b. Tidak Tahu
- Apakah anda mengetahui permainan labirin (*maze game*) ?  
☒ Iya, Tahu      b. Tidak Tahu
- Apakah anda pernah bermain permainan labirin ?  
☒ Iya, Pernah      b. Tidak Pernah
- Apakah anda pernah bermain permainan labirin berbasis *Virtual Reality* ?  
 a. Iya, Pernah      ☒ Tidak Pernah  
 Jika Pernah, bagaimana pendapat anda?
- Apakah anda mengetahui *Artificial Intelligence* ?  
☒ Iya, Pernah      b. Tidak Tahu

### B. PENILAIAN TERHADAP APLIKASI

Isilah pertanyaan di bawah ini dengan menggunakan tanda centang (✓)

Keterangan							
SS = Sangat setuju		S = Setuju		CS = Cukup setuju		STS = Sangat tidak setuju	
KS = Kurang setuju		TS = Tidak Setuju					
No	Parameter Antar Muka	STS	TS	KS	CS	S	SS
1.	Aplikasi memiliki tampilan, warna, dan desain yang menarik.						✓
2.	Aplikasi ini memiliki tata letak tombol, instruksi dan informasi yang mudah dipahami						✓
No	Parameter Immersive	STS	TS	KS	CS	S	SS
3.	Anda merasakan sensasi nyata seperti di dalam <i>maze</i> sungguhan						✓
4.	Anda merasa tertantang untuk segera mencari jalan keluar <i>maze</i>					✓	
5.	Anda merasakan suasana misterius yang ditawarkan oleh permainan					✓	
No	Parameter Artificial Intelligence	STS	TS	KS	CS	S	SS
6.	Musuh dalam permainan ini dapat menemukan anda					✓	
7.	Semakin tinggi level, maka semakin cepat musuh menemukan anda						✓
8.	Tingkat kesulitan musuh sesuai dengan level permainan					✓	
No	Parameter Minigame	STS	TS	KS	CS	S	SS
9.	Minigame Legot dapat melatih kemampuan logika						✓

	anda						
10	Minigame Legot pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai					✓	
11	Minigame Legot pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai						✓
12	Minigame Legot pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai						✓
13	Minigame Picture Memory dapat melatih memori anda						✓
14	Minigame Picture Memory pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai						✓
15	Minigame Picture Memory pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓	
16	Minigame Picture Memory pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai					✓	
17	Minigame Tap Fast dapat melatih kecepatan refleksi anda						✓
18	Minigame Tap Fast pada level <i>easy</i> memiliki tingkat kesulitan yang sesuai						✓
19	Minigame Tap Fast pada level <i>medium</i> memiliki tingkat kesulitan yang sesuai					✓	
20	Minigame Tap Fast pada level <i>hard</i> memiliki tingkat kesulitan yang sesuai						✓
No	Parameter Gameplay	STS	TS	KS	CS	S	SS
21	Rancangan <i>random maze</i> sudah sesuai dengan tingkat kesulitan levelnya						✓
22	Saya merasa <i>game</i> ini cocok dimainkan secara <i>multiplayer</i>						✓
23	Saya merasa dengan bantuan <i>power up</i> , <i>game</i> ini menjadi menarik						✓
No	Parameter Kenyamanan	STS	TS	KS	CS	S	SS
24	Aplikasi dapat berjalan lancar tanpa <i>lag</i> dan <i>crash</i>						✓
25	Kontrol pergerakan <i>player</i> tidak membingungkan					✓	
26	Saya merasa nyaman selama bermain permainan ini					✓	

## C. KRITIK DAN SARAN

Gameplay masih kurang bagus, tetapi kalau bisa  
 hasil untuk peningkatan game nya di perbagus lagi  
 agar pemain lebih menyukai bermain game ini  
 merasa nyaman

*[Halaman ini sengaja dikosongkan]*

## BIODATA PENULIS



Penulis lahir di Pati, 24 September 1996, merupakan anak pertama dari 2 bersaudara. Dalam menjalani pendidikan semasa hidup, penulis menempuh pendidikan di TK Darmawanita, SDN 01 Raci, SMPN 1 Juwana, SMAN 1 Pati, dan S1 Departemen Informatika Institut Teknologi Sepuluh Nopember(ITS) pada rumpun Interaksi Grafika dan Seni.

Selama menjadi mahasiswa, penulis ikut dalam Himpunan Mahasiswa Teknik Komputer Informatika ITS, di departemen Hublu pada tahun kedua, dan dilanjut sebagai staf ahli di departemen Hublu pada tahun ketiga, staf Schematics 2015 dan 2016 di biro Keamanan dan Perizinan.

*[Halaman ini sengaja dikosongkan]*